DATA TRANSFER STRATEGIES FOR OVERSET AND HYBRID COMPUTATIONAL METHODS

A Dissertation
Presented to
The Academic Faculty

by

Eliot W. Quon

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
December 2014

DATA TRANSFER STRATEGIES FOR OVERSET AND HYBRID COMPUTATIONAL METHODS

Approved by:

Professor Marilyn Smith, Advisor,
Committee Chair
School of Aerospace Engineering
Georgia Institute of Technology

Professor Mark Costello
School of Aerospace Engineering
Georgia Institute of Technology

Professor Stephen Ruffin
School of Aerospace Engineering
Georgia Institute of Technology

Professor Eric Johnson
School of Aerospace Engineering
Georgia Institute of Technology

Professor Haomin Zhou
School of Mathematics
Georgia Institute of Technology

Date Approved:

To my parents, Steven and Karen

ACKNOWLEDGEMENTS

First off, I would like to thank my advisor, Prof. Marilyn J. Smith for her many years of support and guidance. She has constantly challenged me and given me the tools and opportunities to succeed in my future career as a researcher. Looking back over the last few years, it is incredible how much I have learned and how quickly the time flies. I would also like to thank the rest of my committee for their guidance and for being so accommodating to my schedule.

The work in this thesis was funded by the Office of Naval Research (under the research grant "High-Accurate Physics-Based Wake Simulation Techniques," ONR N00014-10-C-0190) and the Department of Energy (SBIR Award Number DE-SC0004403). Computing resources were provided by the Department of Defense High Performance Computing (HPC) Modernization Program. Additional support was generously provided by the Achievement Rewards for College Scientists (ARCS) Foundation and the Vertical Flight Foundation.

I would like to acknowledge other mentors that have been a part of my academic and professional growth. Dr. William Warmbrodt at NASA Ames, I would like to thank you for laughing on the phone when I told you that I knew all about CFD, and then bringing me onboard for a summer internship. Thanks to Ethan Romander for actually teaching me about CFD during that internship. I would also like to thank Dr. Pawel Chwalowski at NASA Langley for a more recent experience that allowed me to broaden my understanding about the capabilities and limitations of CFD, and for the opportunity to crawl through the Transonic Dynamics Tunnel. Finally, I would like to thank Dr. Glen Whitehouse for the opportunity to use his VorTran-M wake analysis and for the many years of productive research collaborations.

To all the members of the NCAEL, past and current, thanks for all the good conversations (academic and otherwise), television show references, puns, Risk, coffee and tea alliances, afternoon yoga... we've got everything we need. I would like to especially call out Dr. Nicolas "Stic" Reveles and Dr. Rajiv "Jobin" Shenoy. This graduate experience would not have been the same without you. A special thanks goes out to my "twin," Dr. Zarrin Chua, who has been an incredible friend to me through the thick and thin. Also special thanks to Dr. Alex Coso and the future Dr. David Torello, who have been great friends and supporters over the years. In addition, future Dr. Toby Xu, for feeding me many delicious foods.

Finally, thank you to all my friends and family. Most importantly, to my parents who have always been my biggest supporters, thank you for all your sacrifices so that I might be able to follow my dreams. And thank you for following me around the country as I pursue them. In addition, I would like to thank all the friends I have made here in Atlanta, most notably the Mobile Social crew who showed me how to live and love ATL. Last but not least, I would like to thank Ann Shen who has been so wonderfully patient and supportive, and has kept me grounded throughout this final stretch of graduate school. I could not have done it without y'all.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CFD Computational fluid dynamics.

CFL Courant number corresponding to the Courant-Friedrichs-Lewy condition.

DCI Domain Connectivity Information.

DNS Direct Numerical Simulation.

FMM Fast Multipole Method.

HAWT Horizontal Axis Wind Turbine.

HRLES Hybrid RANS-LES turbulence model.

IDW Inverse Distance Weighting method.

IIM Inverse Isoparametric Mapping.

LES Large Eddy Simulation.

NREL National Renewable Energy Laboratory.

NSWCCD Naval Surface Warfare Center Carderock Division.

PDE Partial differential equation.

RBF Radial Basis Function.

RFB Root Flap Bending.

SKR Steering Kernel Regression.

SRBF Steered Radial Basis Function.

SVD Singular Value Decomposition.

TPS Thin-Plate Splines, $\phi(r) = r^2 \log(r)$.

URANS Unsteady Reynolds-Averaged Navier-Stokes CFD.

V-V Vorticity-velocity formulation of the Navier-Stokes equations.

W2 Compact Wendland C^2 function, $\phi(r) = (1-r)^4(4r+1)$.

NOMENCLATURE

α	Angle of attack, deg.
α_i	Interpolation coefficient for basis function centered at location i
β	Polynomial interpolation coefficient
Δs	Uniform grid spacing
Δt	Simulation time step size
δ_{ij}	Kronecker delta
ℓ	Characteristic length
μ_t	Eddy viscosity
ω	Vorticity
Φ	Isotropic kernel, $\Phi(\vec{x}, \vec{x}_i) = \phi(\|\|\vec{x} - \vec{x}_i\|\|)$
ϕ	Radial basis function (RBF), $\phi(r)$
ρ	Density, kg/m^3
a	Speed of sound, m/s
C_V	Heat capacity at constant volume
E	Total energy, $E = C_V T + u_k u_k/2$
K	Kernel function, $K(\vec{x}, \vec{y})$
k	Turbulent kinetic energy
M	Mach number, $M = V/a$

N	Number of interpolation source points
p	Pressure, N/m^2
Q	Torque, N·m
R	Rotor blade radius, m
r	Radial or Euclidean distance, $\|\|\vec{x}\|\|$
T	Temperature, °K
T	Thrust, N
u	x-velocity, m/s
v	y-velocity, m/s
V_∞	Free-stream velocity, m/s
w	z-velocity, m/s

SUMMARY

Modern computational science permits the accurate solution of nonlinear partial differential equations (PDEs) on overlapping computational domains, known as an *overset* approach. The complex grid interconnectivity inherent in the overset method can introduce errors in the solution through "orphan" points, i.e., grid points for which reliable solution donor points cannot be located. For this reason, a variety of data transfer strategies based on scattered data interpolation techniques have been assessed, with application to both overset and hybrid methodologies. Scattered data approaches are attractive because they are decoupled from solver type and topology, and may be readily applied within existing methodologies. In addition to standard radial basis function (RBF) interpolation, a novel *steered radial basis function* (SRBF) interpolation technique has been developed to introduce data adaptivity into the data transfer algorithm. All techniques were assessed by interpolating both continuous and discontinuous analytical test functions. For discontinuous functions, SRBF interpolation was able to maintain solution gradients with the steering technique being the scattered-data analog of a slope limiter. In comparison with linear mappings, the higher-order approaches were able to more accurately preserve flow physics for arbitrary grid configurations.

Overset validation test cases included an inviscid convecting vortex, a shock tube, and a turbulent ship airwake. These were studied within unsteady Reynolds-Averaged Navier-Stokes (URANS) simulations to determine quantitative and qualitative improvements when applying RBF interpolation over current methods. The convecting vortex was also analyzed on a grid configuration which contained orphan points under

the state-of-the-art overset paradigm. This was successfully solved by the RBF-based algorithm, which effectively eliminated orphans by enabling high-order extrapolation. Order-of-magnitude reductions in error compared to the exact vortex solution were observed. In addition, transient conservation errors that persisted in the original overset methodology were eliminated by the RBF approach. To assess the effect of advanced mapping techniques on the fidelity of a moving grid simulation, RBF interpolation was applied to a hybrid simulation of an isolated wind turbine rotor. The resulting blade pressure distributions were comparable to a rotor simulation with refined near-body grids.

CHAPTER I

INTRODUCTION

Computational science comprises a vital part of every technical discipline, driving the understanding of complicated physical phenomena and the design of complex systems. This field has widely become accepted as an essential part of scientific inquiry to the extent that a Presidential Advisory Committee has identified it as an area of strategic importance requiring increasing levels of long-term funding and support [1]. These resources will permit researchers to study problems of increasing complexity and importance. Within the aerospace community, some sample problems of interest include rotorcraft wakes and maneuvers, wind turbines in typical wind farm operating conditions, aircraft systems, launch vehicles, and payload deployment.

In both rotorcraft and wind turbines, lifting bodies (e.g., the rotor blades) produce vortical wakes (Fig. 1) that interact with other bodies immersed in the same fluid, and the interaction is complex when the wake persists near to the rotor for long periods. For a helicopter, the proximity of the wake induces an inflow that reduces thrust and may cause detrimental acoustic interactions and handling characteristics. As an example, blade-vortex interactions (BVI) may occur between a rotor blade and the tip vortex trailed from the blade that previously swept through the same area. Tip vortices may also impinge on the empennage and tail sections of an aircraft which affect the design of control systems. More unique issues include asymmetric rotor loads in forward flight that result in dynamic stall over a significant portion of the retreating side of the rotor disc. These examples represent a subset of aerodynamics, namely rotary wing aerodynamics, that may be studied with state-of-the-art computational tools.

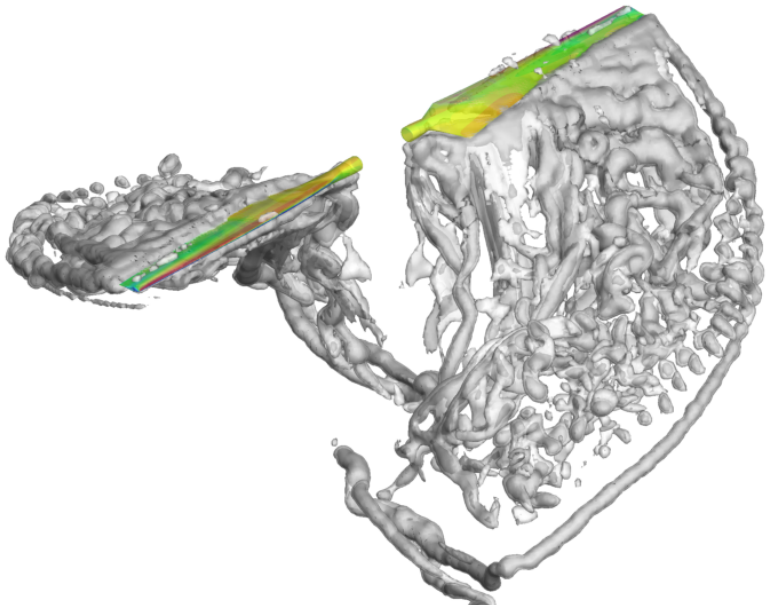


Figure 1: Iso-contours of vorticity magnitude for an isolated wind turbine rotor, computed with a hybrid computational fluid dynamics methodology.

Modern computational science in general permits the accurate solution of nonlinear partial differential equations (PDE) on computational meshes. However, numerical predictions of rotor near field and far field flow phenomena typically involve a trade-off between accuracy and computational expense [2]. Individual wind turbine and wind farm simulations pose additional challenges when accounting for additional couplings between unsteady aerodynamics, atmospheric dynamics, and structural dynamics. Offshore wind turbines may be installed on non-stationary tethered platforms, introducing coupling to hydrodynamics. Similar challenges also occur when simulating aircraft operations in the vicinity of naval ships. The multidisciplinary nature of these problems motivates increasingly sophisticated analysis techniques that in turn present additional computational challenges. For example, a researcher must combine analyses with disparate spatial and temporal scales (e.g., in aeroelastic analyses), synthesize data from deterministic and stochastic models (e.g., in turbulence modeling), and separate numerical error from model limitations. These challenges

may be addressed from two different directions. From a modeling perspective, performing a coupled solution with separate solvers can provide increased flexibility, efficiency, and accuracy. The following sections will introduce two modular methodologies that form the basis for a variety of computational problems. Alternatively, from a fundamental mathematical perspective the exchange of information both within a single code and between multiple codes will generally depend upon a data transfer technique (i.e., interpolation, extrapolation, or regression) to provide information to advance the solution. The subsequent section describes a variety of approaches to transferring data within a variety of computational configurations.

1.1 The Overset Technique

The aforementioned applications of interest often involve predicting the unsteady aerodynamics of both lifting and bluff bodies moving in multiple frames of reference. To facilitate the engineering analysis, an efficient means of handling the evolution of computational domains due to mesh motion, deformation, and/or grid adaptation is necessary. The state of the art in computational fluid dynamics (CFD) is to use an *overset* or *Chimera* approach [3, 4], which applies overlapping grids for the time-accurate solution of unsteady problems and the modeling of complex geometries. This is a modular approach utilizing multiple body-fitted grids to model each moving component, in addition to one or more stationary background grids to model the remainder of the flow field. Overset grids eliminate excessively distorted structured meshes (e.g., near concavities or protrusions in geometry) and unnecessary grid clustering (e.g., in the far field of an airfoil with an H-grid topology). In addition, overset grid systems permit interior grid boundaries to be placed arbitrarily so that components moving in different frames of reference can be simulated. The scheme has since been applied to both structured and unstructured grids for many engineering problems of interest [5, 6, 7, 8].

An overset method requires that flow-field data be exchanged between pairs of overlapping meshes to enable a solution on each component grid so that the global solution is consistent, convergent, and stable. Additional effort is needed to obtain a solution because of the potentially complex domain interconnectivity between multiple overlapped grids. Moreover, since points from neighboring grids are in general non-coincident, interpolation is required for the data transfer at each time step.

1.1.1 Procedure

Prior to flow solution, the Chimera approach requires domain decomposition and mesh interpolation to initiate the data transfer between the overset grids. First, hole cutting is performed to eliminate points that lie outside of the computational domain such as points within solid boundaries. Next, all points adjacent to these holes are marked as fringe points. Any boundary points overset on another mesh are also identified as fringe points. As a result of this procedure, an *iblank* array is passed to the flow solver that identifies which nodes should be solved (known as "in" points), ignored ("out" points), or require data from another grid (fringe points). A search is then performed to identify the nodes from which interpolated data are needed. Fringe points (also known as *receptors* or "ghost" points) are matched with nearby nodes (known as *donors*) on which the interpolated solution will be based. Donors may come from one or multiple grids. From this donor-receptor search, sets of connected points known as *stencils* are typically formed and interpolation weights are calculated. The search operation and calculation of interpolation weights are typically performed by additional software, such as `PEGASUS 5` [9], `PUNDIT` [10], or `Suggar++` [11, 12].

The result of this process is illustrated in Fig. 2, where grids $\mathcal{G}_1$ and $\mathcal{G}_2$ have been overset to create a single composite domain, and a roughly circular set of points has been removed from grid $\mathcal{G}_1$ to accommodate grid $\mathcal{G}_2$. Receptor points have the color of the donor grid, i.e., the grid from which information is received. The light-green

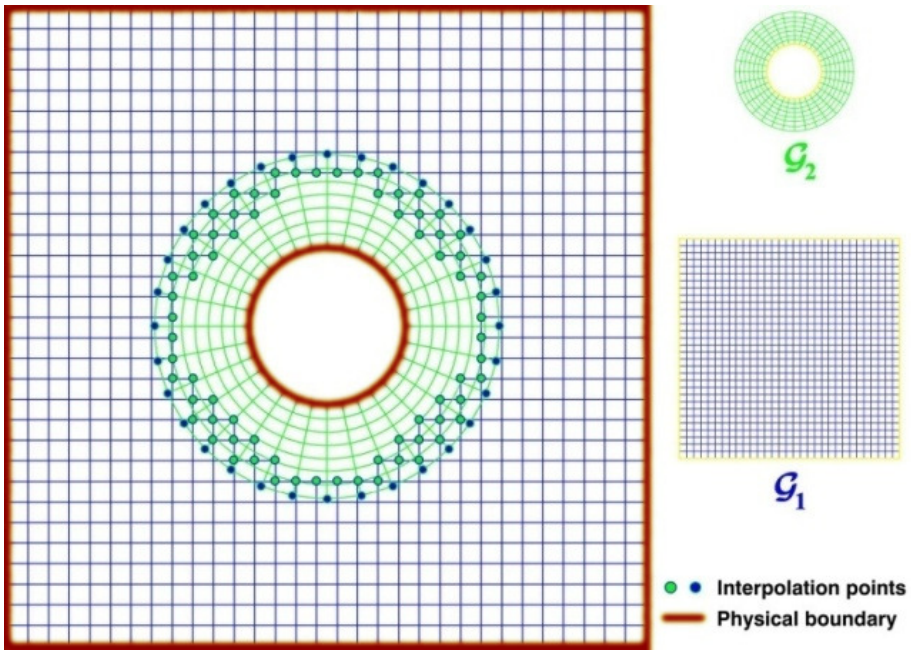


Figure 2: Example of a simple overset grid system [13].

circles are the receptor points from background grid $\mathcal{G}_1$ (adjacent to the hole) that depend on the solution from near-body grid $\mathcal{G}_2$. Similarly, since the domain of $\mathcal{G}_2$ lies entirely within $\mathcal{G}_1$, the filled dark-blue circles indicate receptors in near-body grid $\mathcal{G}_2$ which require donors from background grid $\mathcal{G}_1$. The donor points in all cases are the nodes of the cell (from the donor grid) that contains the receptor point.

After connectivity information for the fringe points on each grid is gathered, the solution at each overset boundary is required for the flow to be resolved on the interior of each grid. On a Cartesian or structured mesh, the most efficient approach is to apply trilinear interpolation [14, 15] (discussed in Section 3.1.2). This is the approach employed by the `PEGASUS 5` grid preprocessor [9] that produces a continuous interpolated function with discontinuous gradients. A more general approach relies on isoparametric mappings (discussed in Section 3.1.3) with trilinear basis functions that are applicable to both structured and unstructured grids. This is the technique employed by `PUNDIT` [10] and `Suggar++` [16].

1.1.2 Orphan Points

Complications arise when suitable points for interpolation (donor points) cannot be found, giving rise to "orphan" points. This situation occurs if adjacent grids have insufficient overlap or if significant disparities in mesh spacing between grid levels exist. For example, a point is considered to be an orphan when its donors are fringe points (that also require an interpolated solution) or out points (outside the computational domain). Two levels of fringe points are required on each grid to preserve high-order accuracy [17, 18]. When orphan points are present, solution fidelity may be lost because two levels of fringes cannot be resolved. Interpolation accuracy is also reduced because fewer points are available from which to perform the data transfer. The problem of orphans is exacerbated by relative mesh motion which can increase the number of orphans and/or change their locations over time. The treatment of orphans may be tailored to a particular flow solver or application, but typically values are estimated at orphan locations with an averaging procedure [19, 20, 8].

When orphan points are present in a solution, two general mitigation approaches exist. First, the grids may be redesigned to improve the quality of mesh intersections. In a recent application of an unstructured near-body methodology coupled to a Cartesian off-body solver, Abras and Hariharan [21] had to manually adjust the trim distance dictating the amount of overlap between near-body and off-body meshes. However, straightforward adjustments are not always possible, especially when considering complex geometries, and grid refinement can significantly increase cost. For example, a wing-store configuration studied by Power et al. [8] had 0.5% of all cells orphaned; application of an adaptive mesh refinement procedure was able to eliminate all orphans but increased the total cell count by 10%. Even if increased mesh sizes are acceptable, it may be difficult to guarantee that meshes in relative motion will be orphan-free for all time steps throughout the simulation. As an alternative, a dense interface grid may be added in the orphan region [22, 8]. Both

approaches require user intervention and added cost, either in engineering hours or computational time.

1.1.3 Conservation Considerations

A well-debated topic in overset CFD is whether data transfers should be performed in a manner that explicitly satisfies conservation laws. The importance of conservation was noted by Steger [23] over two decades ago, and a consensus has not been reached within the overset community. In general, conservation within the context of a fluid volume refers to *mass conservation*, where *mass* may be defined as:

$$m = \int_{\mathcal{H}} u \, \mathrm{d}\Omega \tag{1}$$

for a mesh $\mathcal{H}$ within a computational domain Ω [24]. The concept of mass may refer to any integrated quantity; for a finite-volume scheme, this is analogous to the conserved quantities of density, momentum, and energy. Conservation is then by definition:

$$\int_{\mathcal{H}^1} u \, \mathrm{d}\Omega = \int_{\mathcal{H}^2} \tilde{u} \, \mathrm{d}\Omega \tag{2}$$

for an interpolation of u on mesh $\mathcal{H}^1$ to a differing mesh $\mathcal{H}^2$. These interpolated values are written as $\tilde{u}$. An interpolation scheme is considered conservative if the numerical approximation to the continuous integral of a dependent variable is equal for adjacent cells [25]. From another perspective, enforcing conservation ensures that the original and interpolated values satisfy the governing conservation laws (e.g., the Navier-Stokes equations).

Non-conservative data transfer represents a physical inconsistency, but the significance of these errors are typically difficult to quantify [23]. Chesshire and Henshaw [26] observed that standard linear interpolation is sufficient to transfer variables that vary smoothly, but more recently Wang et al. [27] have demonstrated that conservative schemes can improve solution accuracy even for smooth flows. As an added benefit, Tang et al. [28] observed that mass-conservative schemes also tended to accelerate convergence.

Conservation errors are most apparent in discontinuous fields. Tang and Zhou [29] studied an inviscid shock solution and observed errors in shock location and structure when conservation was not maintained. The shock speed was also 8% faster than the exact solution, although all of these errors could be mitigated through mesh refinement. Other authors have described similar results. Chesshire and Henshaw [30] reported that conservative interpolation is necessary to capture the speed and location of shocks when modeled by Burger's Equation on computational grids that are not sufficiently refined. Zheng and Liou [31] similarly reported that a non-conservative Chimera scheme tended to over-predict shock speed in a benchmark shock tube problem calculated by a Navier-Stokes methodology. Pärt-Enander and Sjogreen [25] demonstrated that a slow-traveling shock wave can have difficulty passing through a non-conservative overset interface, either slowing down or stopping at the boundary. However, enforcement of conservative flux interpolation was not straightforward and resulted in spurious oscillations at the overset interface if both inflow and outflow characteristics were considered. Improved interpolation methods were developed by Pärt-Enander and Sjogreen [25] whereby solution stability was maintained by a characteristic decomposition and application of a nonlinear filter.

Issues with conservation may be mitigated by grid refinement and/or increased number of iterations. Wang et al. [27] showed that as a given grid is refined, both non-conservative and conservative schemes will converge to the same solution. This refinement may also be accomplished by local grid adaption [15]. However, the assumption that the mesh will be sufficiently fine in the overlap region is not always valid, especially for dynamic cases. As an alternate strategy, Freitas and Runnels [32] circumvented mass-conservation issues at additional cost by iterating until residuals were sufficiently small, where the definition of small is problem-dependent.

1.2 Hybrid Techniques

In a broad sense, hybrid approaches are a variation on the traditional overset methodology with the key distinction being that the overset computational domains are treated by different solvers. Each solver may be based on different physical models with no spatial or temporal dependence on each other. While an overset approach is modular with respect to modeled geometry, a hybrid approach may be thought of as being modular with respect to modeled and resolved physics. Solution quality may be enhanced by selecting methods that emphasize certain physics relative to the problem under investigation. From an engineering perspective, obtaining a coupled solution with different solvers resolving different levels of governing equation fidelity is often more attractive than using a single integrator. When sets of governing equations are solved separately, the choice of numerical method and implementation are more straightforward. Using this approach, substantial improvements in terms of both spatial accuracy and solver efficiency have been recently achieved for both lifting and bluff bodies [33, 34, 35].

When considering rotating system aerodynamics, early hybrid approaches have represented the trailed vorticity in the wake of the rotor using vortex filaments whose fundamental solution may be derived from potential flow fluid dynamics. These filaments are efficient representations of the flow-field structure and satisfy the Helmholtz theorems: 1) The strength of a vortex filament is constant over its length; 2) a vortex filament must form closed loops, terminating at a solid boundary or extending to infinity; and 3) a fluid that is initially irrotational will remain irrotational unless acted on by shear forces. Since the wake is no longer solved by a set of PDEs, it is no longer necessary to use an Eulerian representation with discrete fixed points in space. With appropriate numerical methods, the cost of evolving and evaluating the wake may be several orders of magnitude less than the cost of unsteady Reynolds-Averaged Navier-Stokes (URANS) methodologies typically employed to calculate the near-body

flow. The efficiency gains are achieved by eliminating the far-field background mesh or drastically reducing its extent, thus reducing the number of points handled by the solver [34, 6]. Horn et al. [36] were able to develop a physically accurate, real-time flight simulation. D'Andrea [37] developed an unstructured panel analysis coupled to a free-wake constant vorticity contour method for the simulation of rotorcraft and tiltrotors. This code was parallelized with a multiblock scheme enabling over an order of magnitude cost reduction over conventional URANS methodologies. Other recent potential-based hybrid methods include a thin-layer Navier-Stokes solver coupled to a free-wake methodology by Sitaraman and Baeder [38]. Overall, this hybrid approach was nearly six times faster because of a reduction in mesh size of approximately the same factor. While this analysis was comparable to the same standalone thin-layer URANS methodology for high-speed flight, greater variations in airloads were observed for cases involving stall.

These approaches become less tractable when modeling the next generation of aerospace technologies. A helicopter in very high-speed forward flight will encounter significant regions of stalled and/or reversed flows on the retreating side of its rotor. In this situation vortex filaments become distorted and the analysis relies on empirical models [39]. In addition, a persistent issue with hybrid methods is that pitching moment predictions are generally poor as noted by Wachspress et al. [34]. A review of international efforts to study the HART-II rotor conducted by Smith et al. [40] also found that hybrid methodologies tend to produce errors in pitching moment magnitude and phase. Moreover, the hybrid methods showed greater sensitivity to blade vortex interaction (BVI) events than the conventional methods.

1.3 Data Transfer Techniques

Linear interpolation is heavily utilized within many fields of engineering and is a necessary component of both the overset and hybrid methodologies. For the purposes of

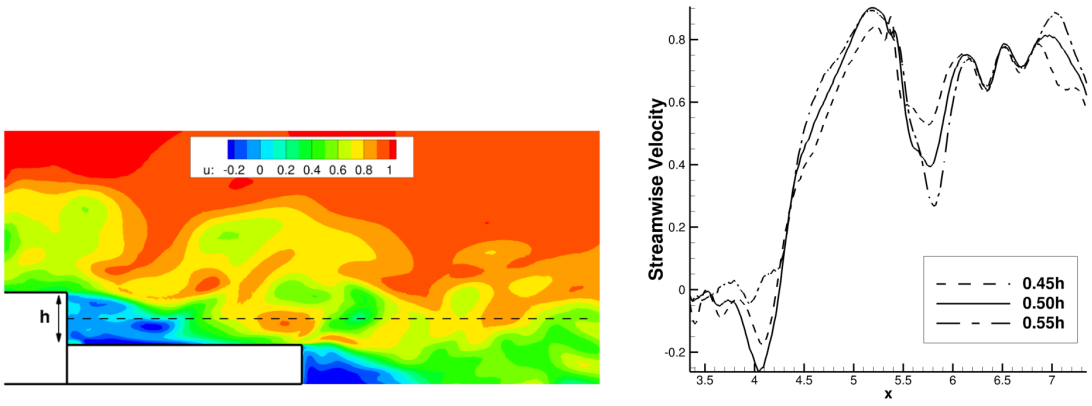
solution transfer between computational meshes and/or solvers, it is standard practice on structured meshes to use trilinear interpolation [41], equivalent to performing a series of one-dimensional first-order polynomial interpolations [14]. Similarly, linear finite element mappings may be applied to unstructured meshes. These approaches, while only second-order accurate [15], are attractive for their simplicity in implementation, low cost, and monotonicity in behavior. In addition, linear approaches also guarantee that the interpolated solution will not exhibit overshoot or undershoot between known points, namely that they satisfy the *maximum principle* [24]. Within the context of solving PDEs, these methods provide non-oscillatory, stable solutions at the expense of numerical diffusion. If purely first-order schemes are used, many problems of interest may become intractable due to the number of computational points required. Higher-order mapping methods are thus more attractive because they have the ability to resolve solutions using much fewer computational points. Since computational meshes are typically designed with higher-order spatial solution schemes in mind, it is not feasible to assume that the flow-field data can be sampled at spacings sufficiently small to guarantee adequate data reconstruction using linear interpolation techniques.

1.3.1 Mathematical Inconsistency

A fundamental issue may exist in the inconsistency between the order of a numerical discretization scheme versus the order of the interpolation scheme applied to transfer solutions between computational domains. Many flow solvers, especially those based on a structured grid topology, have a spatial accuracy that is third-order or higher [42, 43]. In comparison, the standard interpolation method is trilinear interpolation, which only achieves formal second-order accuracy [15]. Linear interpolation does not always accurately characterize mean flows and recent work has demonstrated overset interpolation-induced distortion of flow features when using linear

techniques [44]. A well-understood numerical challenge is the adequate resolution of transonic and supersonic flows, which are characterized by jump discontinuities (shocks) that traditional interpolation techniques tend to dissipate. Turbulent fields are stochastic and also discontinuous in nature and thus present similar numerical challenges.

For example, flow over a model naval frigate approximates a backward-facing step in cross-section (depicted in Fig. 3a). To illustrate the behavior of turbulent flow quantities, the streamwise velocity has been sampled at half the hangar height h, and at $50\%h \pm 5\%h$. In this example, the discontinuous nature of the flow accounts for spatial velocity variations that are over 20% of the free-stream velocity in magnitude, occurring over a distance of only 5% of the hangar height (Fig. 3b). For all applications of interest, the preservation of nonlinear flow features across overlapped grid regions is essential. Development of more robust approaches to solution transfer capable of high accuracy in the presence of both smooth and discontinuous flow features is needed to strengthen future overset analyses.



(a) Model cross-section with streamwise velocity contours (b) Sampled velocities along dashed line

Figure 3: Flow sampled in the wake of a naval frigate.

1.3.2 Application to Overset and Hybrid Schemes

While overset interpolations are often based on linear algorithms, there exists motivation to employ higher-order schemes. The most important reason is to maintain a level of interpolation accuracy that is consistent with the discretization accuracy. Chesshire and Henshaw [26] proved that in order to maintain discretization order accuracy in a solution, the order of the interpolation scheme should be at least the same order as the difference scheme, if not higher, depending on the size of the overlap region. Similarly, Sherer and Scott [45] proved through a Fourier analysis that low-order interpolation may reduce the overall accuracy of a solution obtained with a high-order numerical method. Recently, Foster and Noack [44] demonstrated that the vortical wake from a cylinder in crossflow resulted in a 4% elongation of shed vortex structures when convected through an overset patch. For applications that require extremely high resolvability (e.g., when considering acoustic wave propagation), Delfs [46] demonstrated that second-order bilinear interpolation error deteriorated the global solution. For the same problem, a mixed fourth- and fifth-order scheme that accounted for solution gradients was found to be stable.

Aside from increasing the order of polynomial interpolation, a number of alternative techniques exist, and a paradigm shift in interpolation techniques may prove to be more effective. The well-known class of radial basis functions (RBFs) has been shown to be effective at reconstructing scattered data and can be readily extended to any dimension [47]. This approach does not require an interpolation stencil of any particular form to determine interpolation weights, so it can operate on arbitrary clouds of data points; a thorough discussion is presented in Section 3.1.5.

1.4 Thesis Objectives

The objective of this thesis is to develop data transfer and solution strategies that enable and facilitate the solution of the next generation of computational problems.

These strategies advance the state of the art in both overset and hybrid computational methodologies. The applicability and value of advanced mapping techniques over traditional linear approaches will be demonstrated wherever possible.

Previous sections have identified a number of knowledge gaps in the areas of mathematical mapping techniques within the context of computational science, overset simulations, and hybrid methodologies. First of all, the limitations of linear techniques are well understood. While scattered or "cloud"-based data techniques are well-established within other fields (e.g., computer graphics, digital elevation modeling, or optical design) their application to CFD problems has been limited. Secondly, issues surrounding orphan points and non-conservative data transfer are well-understood by the overset community but a robust and cost-effective solution to the problem of insufficient overlap has yet to be presented. Finally, hybrid simulations have room for accuracy improvements when resolving complex, stalled flow fields. Motivated by this knowledge, this thesis aims to advance the computational state of the art by accomplishing the following:

- **Evaluating the impact of different data transfer techniques** on interpolated solution accuracy. This includes investigations of various basis functions, sizes of interpolation clouds, and relevant algorithm parameters. Results are quantified in terms of solution error when an exact solution is available.

- **Reducing or eliminating orphan points** using scattered data techniques, which provide mappings of arbitrarily structured data samples in any dimension. Investigations of data transfer with orphans also assesses the effectiveness of the approach at data extrapolation.

- **Isolating overset errors** from combined spatial and temporal error, which permits accurate assessment of interpolation errors within the overset framework. Results are compared to an analytical solution or corresponding single

grid calculation.

- **Applying scattered data techniques to hybrid methodologies** to more accurately interface URANS and vorticity-velocity (V-V) solvers.

- **Implementing a decoupled boundaries methodology** within the URANS and vorticity-velocity (V-V) solver hybrid framework. This approach enables arbitrary overset or non-contiguous configurations in which component grids may be translated or rotated without any overlap or background grid requirements.

- **Developing a data-adaptive interpolation strategy** that can accurately reproduce both smooth and discontinuous solutions. A steering strategy based on flow-field gradients is applied to introduce anisotropy into basis functions in the presence of solution discontinuities.

To accomplish these objectives, a variety of analytical test functions as well as inviscid and turbulent flow solutions were considered, in addition to the following validation test cases:

- Inviscid convecting vortex

- Shock tube

- Turbulent naval ship airwake

- Horizontal-axis wind turbine (HAWT) under stalled conditions

This contributions of this thesis are summarized by the following research questions:

- How might higher-order scattered data interpolation techniques be applied to improve data transfer accuracy for general computational science problems? Can robust techniques that are capable of representing both smooth and discontinuous flow fields be developed?

- How can these data transfer strategies be applied to overset simulations? More specifically, can these techniques mitigate the effects of orphan points in terms of reduced accuracy and/or increased cost?

- Can a more robust hybrid approach be developed to simulate complex, stalled flow fields? Can this approach produce results comparable to a traditional unsteady Reynolds-Averaged Navier-Stokes (URANS) simulation at reduced cost?

CHAPTER II

METHODOLOGIES

This chapter describes the baseline methodologies employed in this thesis to compute unsteady, three-dimensional flow solutions with complex geometries. The first section provides the governing equations for all simulations in the present work. Then traditional flow simulations and hybrid configurations are discussed. The specific solvers and computational tools that have been employed will be described in the final section.

2.1 The Navier-Stokes Equations

The governing equations in fluid dynamics are:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0, \tag{3}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_i u_j) = \frac{\partial}{\partial x_j}\sigma_{ij}, \tag{4}$$

and

$$\frac{\partial \rho E}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i E) = \frac{\partial}{\partial x_i}\left[-q_i + \sigma_{ij}u_j\right]. \tag{5}$$

These three equations represent mass, momentum, and energy conservation, respectively. The stress tensor, σ_{ij} can be decomposed into normal and shear stress terms:

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij}. \tag{6}$$

For a Newtonian fluid, the shear stress tensor is:

$$\tau_{ij} = 2\mu S_{ij}, \tag{7}$$

with the strain rate tensor defined as $S_{ij} \equiv \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$. The problem is closed by the ideal gas equation of state, $p = \rho RT$ at moderate speeds and temperatures.

Eqns. 3, 4, 6 and 7 are combined to form the Navier-Stokes equations:

$$\rho\frac{\partial u_i}{\partial t} + \rho u_j\frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \mu\frac{\partial^2 u_i}{\partial x_j x_j}. \tag{8}$$

2.1.1 Reynolds-Averaged Navier-Stokes

A Reynolds decomposition is applied to the governing Navier-Stokes equations (Eqn. 8), i.e., $u(t) = u + u'(t)$ where u is a mean flow quantity and $u'(t)$ is a fluctuation. Averaging the result gives:

$$\rho\frac{\partial u_i}{\partial t} + \rho u_j\frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \mu\frac{\partial^2 u_i}{\partial x_j x_j} - \rho\overline{u_i' u_j'}. \tag{9}$$

The term with the averaged product of fluctuations, $\overline{u_i' u_j'}$, is known as the Reynolds stress tensor and permits momentum transport due to velocity fluctuations.

2.1.2 Velocity-Vorticity Form

Vorticity (ω) is related to velocity (u) through the mathematical curl operation, $\boldsymbol{\omega} = \nabla \times \boldsymbol{u}$. The Navier-Stokes equations can be used to describe a vorticity field by taking the curl of Eqn. 9, resulting in the unsteady vorticity transport equation [48]:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{\omega} - \boldsymbol{\omega}\cdot\nabla\boldsymbol{u} = \nu\nabla^2\boldsymbol{\omega} + \boldsymbol{S}. \tag{10}$$

The vorticity source term (S) can arise from any aerodynamic loading model, and the viscosity (ν) is typically neglected. Since a dependence on velocity remains, the problem is closed by the Biot-Savart equation, describing the induced velocity for a given vorticity distribution [48]:

$$\nabla^2\boldsymbol{u} = -\nabla\times\boldsymbol{\omega}. \tag{11}$$

Equation 11 may be efficiently evaluated using a fast multipole method (FMM) [49].

2.2 Traditional URANS CFD

Both engineering and research implementations of computational fluid dynamics (CFD) commonly solve the unsteady Navier-Stokes governing equations for fluid dynamics in Reynolds-Averaged form. Solutions for density, velocities (or momentum), and pressure (or total energy) are obtained from simultaneous solution of the continuity, conservation of momentum, and conservation of energy equations. Internal energy is a function of the temperature throughout the computational domain, thus for problems of interest in this effort air is represented as an ideal gas, and the equation are closed by the ideal gas law.

To resolve the effects of turbulence, the shear stress tensor in the momentum equations (Eqn. 9) is closed by the Boussinesq hypothesis:

$$\rho\overline{u_i'u_j'} = 2\mu_t\left[S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right] - \frac{2}{3}\rho k\delta_{ij}. \tag{12}$$

This approximates the Reynolds stress as a function of eddy viscosity (μ_t), a quantity analogous to molecular viscosity used to calculate turbulent transport and dissipation in URANS simulations. Turbulence modeling with either algebraic or differential equations (of one or more turbulence variables) is required to calculate the evolution of eddy viscosity. Higher-fidelity approaches include Large-Eddy Simulations (LES) and Direct Numerical Simulations (DNS) that resolve the large-scale turbulence or all scales of turbulence, respectively. A hybrid approach is also possible, whereby a blending function is applied to smoothly switch between URANS-modeled turbulent kinetic energy (k) near a body and LES-calculated k at larger scales with sub-grid scale turbulence modeling at smaller scales. This enables a combined RANS/LES solution through the eddy viscosity.

2.3 Hybrid CFD

Successful implementation of a hybrid methodology requires an appropriate grid configuration and interface boundary condition. The former maximizes efficiency and

efficacy of the different solvers by sizing the computational domains of each solver to match the extent of its applicability, while the latter ensures solution integrity between codes. Although URANS methodologies are appropriate for all parts of the flow field, its domain will overlap with far-field regions where wake solvers are also applicable. Therefore the extent of the near-body meshes is typically minimized to allow for more computationally efficient solvers to calculate the larger flow domain.

Different overset grid configurations are compared in Fig. 4. In the traditional setup (a), a large background mesh is required to realistically resolve the flow. Since the solution is governed by PDEs that are elliptic in space, the far-field boundaries should be located sufficiently far from lifting bodies to approximate an unbounded domain. To minimize numerical diffusion, grid points should be strategically clustered in regions of large gradients, necessitating a priori knowledge of the wake structure and/or a mesh adaption strategy.

For hybrid simulations, the extent of the background mesh is greatly reduced, as depicted in Fig. 4(b), and the far-field boundary condition (based on Riemann invariants [50]) is replaced with an interface boundary that accounts for the influence of the wake. The background mesh reduction is less pronounced (for basic grid geometries) when additional bodies are introduced into the simulation. In addition, the overset approach is still applied to preserve the capability of simulating multiple moving bodies and to transfer the flow solution between moving and stationary reference frames. In this approach, the near-body solution is transferred twice, once between overset meshes and once between the interface boundary and the wake code. The additional data transfers increase the opportunity for a loss of fidelity in terms of solution values, gradients, or integrated quantities (e.g., mass).

A simpler and potentially more accurate approach is to remove the intermediate grid, which eliminates one of the solution transfer operations. The coupling between the near-body and wake solvers then occurs through a moving interface boundary on

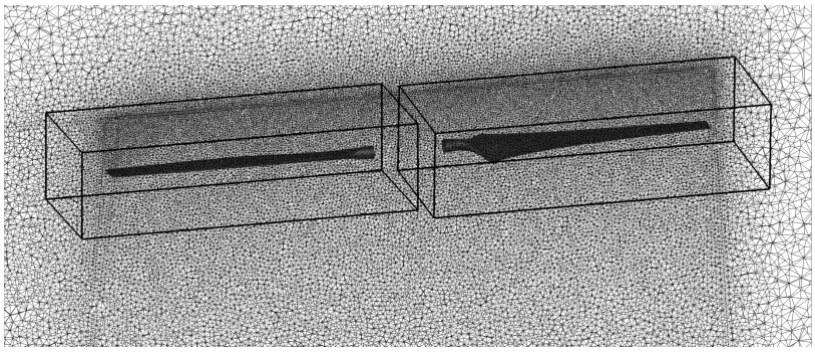
the near-body grid. A very efficient approach for studying rotors in axial flow is to take advantage of axisymmetry (Fig. 4c) and to only simulate a single rotor blade. The solution from the blade is replicated and rotated to other azimuthal locations. This approach is quickly invalidated with yawed flow and/or more complex systems. Therefore a more general approach is to decouple the boundaries (Fig. 4d) within the URANS domain. In this case, all lifting bodies are represented in the near-body domain without any overset meshes (i.e., the grids are non-contiguous) and their solutions are instead coupled through a wake analysis. This approach also eliminates the overset data transfer step and permits the arbitrary placement and relative motion of grids. Further discussion of this methodology is provided in Section 6.1.3.

2.4 Computational Tools

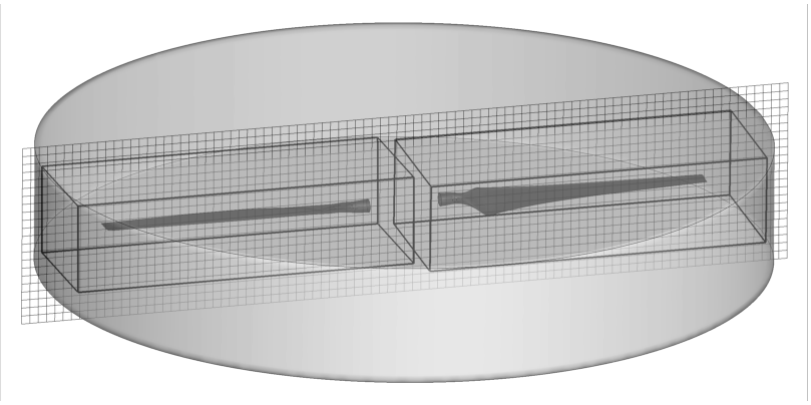
2.4.1 Unsteady RANS: FUN3D

The baseline CFD solver that was selected for overset development is FUN3D [51], a fully unstructured URANS solver developed primarily at NASA Langley Research Center with additions from the Georgia Institute of Technology. Numerical schemes implemented in the code are spatially second-order accurate with Roe upwinding and temporally second-order accurate using multi-step backward difference formulas. FUN3D includes overset mesh capabilities for accurate resolution of complex geometries and multiple frames of motion, and has been applied extensively for a variety of simulations including rotorcraft [52].

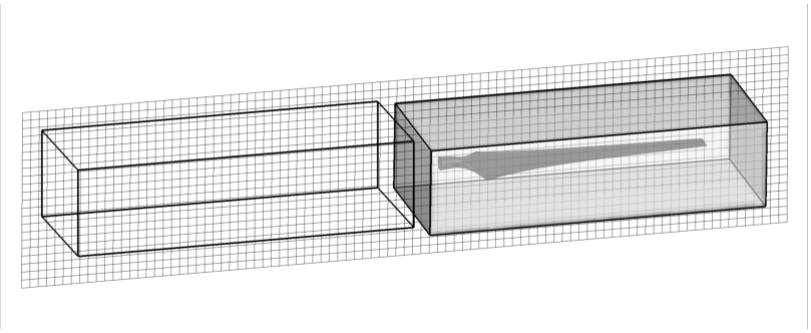
This solver has modeled both complete overset computational domains and near-body regions coupled with other wake analysis codes. An unsteady hybrid Reynolds-averaged Navier-Stokes/Large eddy simulation (HRLES) turbulence model developed at the Georgia Institute of Technology enables large-scale turbulence to be captured throughout the computational domain except at the configuration surface where they are calculated by the Menter k-Ω SST (shear stress transport) model [53].



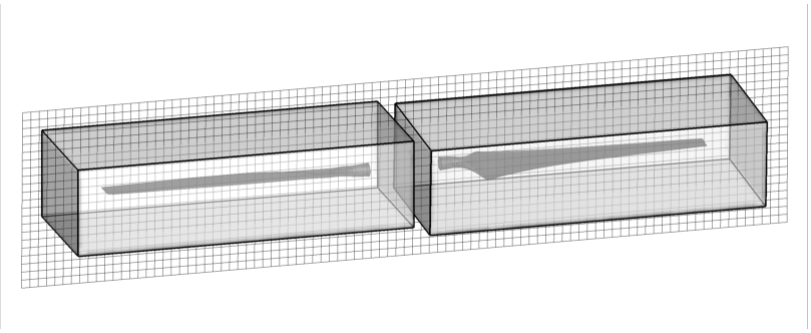
(a) Traditional Overset



(b) Overset Hybrid



(c) Hybrid (Axisymmetric)



(d) Hybrid (Decoupled Boundaries)

Figure 4: Comparison of computational meshes/domains used in four different overset configurations.

2.4.2 Vorticity-Velocity: VorTran-M

For the hybrid analyses in the present work, the far-field computational domains are resolved by VorTran-M [48], a three-dimensional volumetric wake solver developed by Continuum Dynamics Inc. (CDI). VorTran-M performs a DNS solution of the unsteady incompressible Navier-Stokes equations in vorticity-velocity form. Velocities are computed at each step using a Cartesian fast multipole method. A discretization of Eqn. 10 is obtained by Toro's Weighted Average Flux (WAF) algorithm and Strang spatial splitting which explicitly conserves vorticity. The effects of numerical diffusion are controlled by the WAF flux limiter. Solid bodies are not directly modeled but can be represented through solution coupling to a near-body solver. The formulation has demonstrated high-resolution rotor wake predictions on low-cell count grids when coupled to a variety of near-body solvers (Euler, RANS, structured, unstructured, octree Cartesian, and overset) for a number of vorticity-dominated flows [39, 54, 55].

2.4.3 Suggar++/DiRTlib

Two auxiliary codes are used to provide overset capabilities to FUN3D: Suggar++ (Structured, Unstructured Generalized overset Grid AssembleR) and DiRTlib (Donor Interpolation Receptor Transaction library) [19, 56]. Suggar++ handles grid transformations and hole-cutting, donor-receptor identification, and calculation of interpolation weights; DiRTlib interfaces with the flow solver and uses the domain connectivity information generated by Suggar++ to calculate interpolated values. A general isoparametric mapping approach (discussed in Section 3.1.3) with trilinear basis functions is applied to interpolate between structured and unstructured grid topologies. For structured meshes, the mapping is based on quadrilateral and hexahedral elements (in two and three dimensions, respectively) while unstructured meshes apply triangle and tetrahedral elements [16]. For static meshes, Suggar++ performs as a preprocessor to match donor and receptor points, after which interpolation weights

are calculated based on donor locations only. For dynamic meshes, Suggar++ is both a preprocessor and an iterative routine operating within FUN3D. At each solver time step and subiteration for time-accurate simulations, DiRTlib interfaces with the solver to retrieve the solution at donor points and update the solution at fringe points.

CHAPTER III

HIGHER-ORDER DATA TRANSFER METHODS

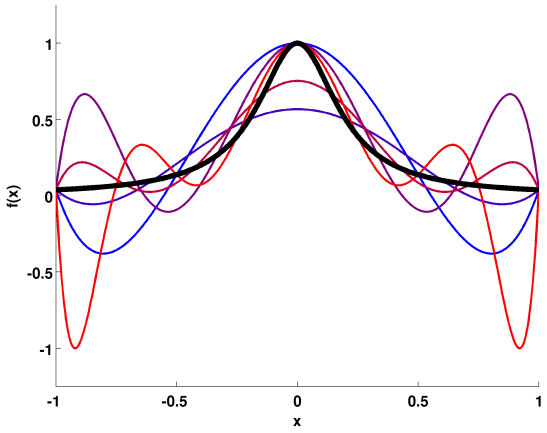
The first section of this chapter provides an overview of established techniques and explores their extension to higher order. Within the framework of polynomial-based interpolation, this involves adding additional points along one dimension. Interpolation with radial basis functions (RBFs), known more broadly as kernel function interpolation [57], is also discussed. This is one possible approach that replaces the polynomial interpolation *stencil* (a set of connected nodes) with an interpolant based on a point *cloud*, a set of points within a neighborhood of the target point with no requirements on interconnectivity. Similar to the choice of interpolation node locations in one dimension, proper selection of source points has been shown to improve solution quality [58]. Throughout this thesis, the terms *kernel* and *basis* (usually radial) are used interchangeably. The second section presents additional justification of the interpolation techniques chosen for evaluation in this thesis. The third section presents two basic studies to evaluate the effectiveness of inverse isoparametric mapping (IIM) when applying higher-order shape functions. Results are compared with the RBF scattered data interpolation technique. In the last section, a thorough investigation of the RBF technique has been conducted with a focus on interpolating quantities relevant to overset and hybrid CFD, addressing interpolant integration to verify conservation losses.

3.1 A Review of Existing Mapping Methods

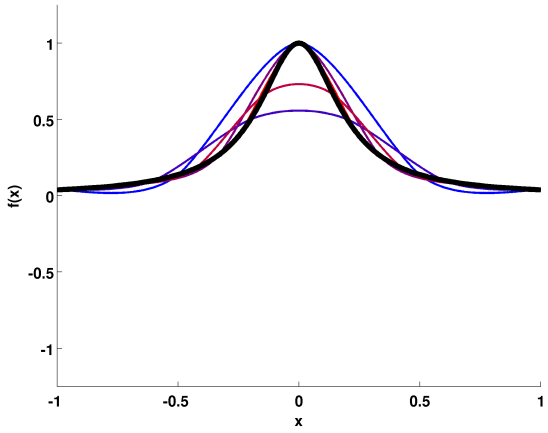
The order of an interpolation method is the order of error term associated with the interpolant based on a measure of the spacing between grid points. The actual observed error is expected to be less than or equal to this order of magnitude. Among the

higher-order approaches, there is no consensus on what approach is optimal. Chaplot et al. [59] present a short summary of recommendations by different authors given different types of sampled data for digital elevation modeling. In general, the method that provides the most accurate reproductions varies depending on the existence of strong spatial structures, variability, and anisotropy of the data. All three of these characteristics are representative of aerodynamic computational flow field variables. While RBF approaches were not necessarily the optimal choice, they were consistently able predict anisotropic data at multiple scales better than kriging or inverse distance weighting approaches.

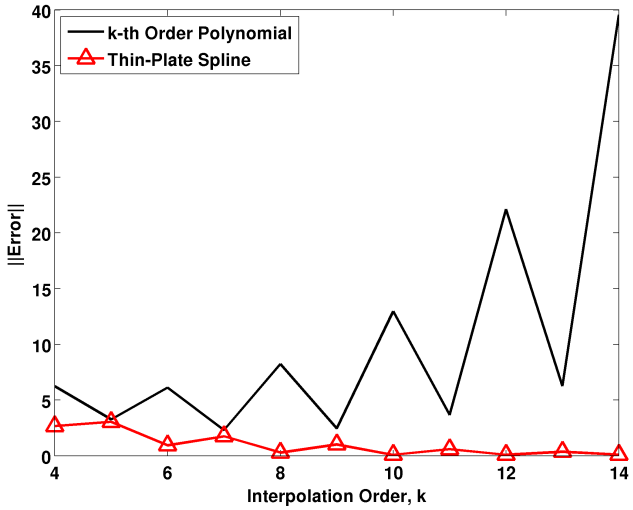
When employing any higher-order approach, additional data must be acquired and utilized judiciously, lest the solution become both less accurate and/or more expensive. In general, higher-order methods require additional operations to incorporate extra information, thus accruing additional computational cost. Moreover, higher-order methods are not necessarily more accurate and may result in non-physical and unstable solutions. This is analogous to the issue of overfitting within the context of statistical models. A common example is the observation that higher-order polynomial interpolation introduces a "ringing" effect into the interpolated solution known as the Runge Phenomenon (Fig. 5a). This has traditionally been mitigated by proper choice of interpolation nodes (e.g., Chebyshev nodes) or with spline interpolation (which applies piecewise-defined polynomials). This has been illustrated by Fig. 5, which demonstrates that straightforward application of polynomial interpolation can result in solution instability (Fig. 5a). Conversely, RBF interpolation with $k + 1$ source points (replicating a k-th order polynomial approach) resulted in a convergent solution (Figs. 5b and 5c). While radial basis function (RBF) interpolants can still exhibit Runge's phenomenon manifested as divergence or slowed convergence [60], scattered data techniques have the potential to be a more general and accurate than linear approaches.



(a) High-order polynomial interpolation



(b) Radial basis function interpolation with thin-plate splines



(c) Error comparison

Figure 5: Runge function, $f(x) = \dfrac{1}{1+25x^2}$, interpolated with equally spaced points to demonstrate Runge's phenomenon. The interpolation order is increased from fourth (blue curve) to eighth order (red curve) in (a) and (b).

A similar and well-understood anomaly is the Gibbs Phenomenon, in which the interpolated solution contains spurious oscillations near discontinuities or high-gradient regions. The undesirable Gibbs phenomenon observed in Fourier series expansions or polynomial interpolation (Fig. 6a) is also found in RBF interpolants as well, as illustrated by Fig. 6b. While the behavior is similar, the mathematical mechanisms in an RBF approach that drive the successive oscillations are currently not well understood. Fornberg and Flyer [61] observed that the decay in RBF interpolants have both exponential and algebraic components as opposed to simple exponential decay associated with traditional techniques. As before with the Runge phenomenon, the scattered data technique is convergent and the amount of overshoot does not increase with the approximation order.

3.1.1 Polynomial Interpolation

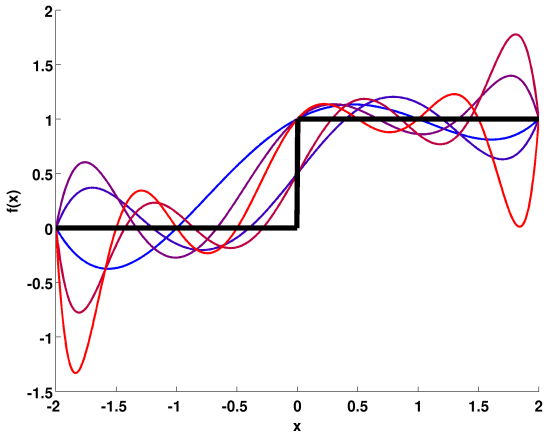
The most common basis function for interpolation is the Lagrange polynomial. Lagrange's formula [14] is:

$$P(x) = \sum_{j=1}^{N} y_j \ell_j(x), \tag{13}$$

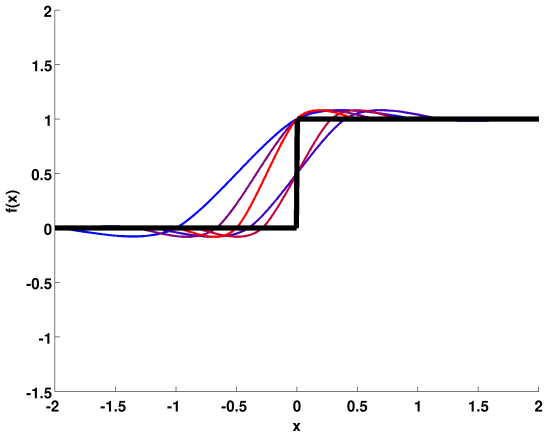
where

$$\ell_j(x) = \prod_{\substack{1 \le i \le N \\ i \neq j}} \frac{x - x_i}{x_j - x_i}. \tag{14}$$

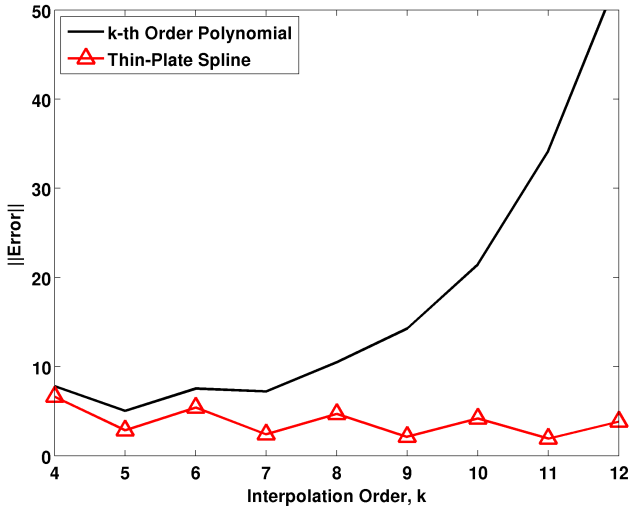
The order (or degree) of the polynomial is $N - 1$, where N is the number of points supporting the interpolation. For small datasets or local subsets of data, increasing the interpolation order more accurately approximates the data at arbitrary locations provided the function is smooth [62]. Polynomial interpolation is reliable only within the interpolation stencil and extrapolation is strongly discouraged [63]. Recent computational methodologies have demonstrated improvements over trilinear interpolation using high-order polynomial interpolants [18, 44]. However, excessively high-order stencils can also cause inaccuracies with spurious oscillations often introduced into the interpolated solution. For example, Desquesnes et al. [64] observed



(a) High-order polynomial interpolation



(b) Radial basis function interpolation with thin-plate splines



(c) Error comparison

Figure 6: A jump discontinuity interpolated with equally spaced points to demonstrate Gibb's phenomenon. The interpolation order is increased from fourth (blue curve) to eighth order (red curve) in (a) and (b).

that errors from fourth-order Lagrange polynomial interpolation give rise to spurious acoustic modes. Lee et al. [65] also applied a fourth-order interpolation approach, based on the finite volume method. A limiter was applied for stability in problems involving a blunt body and a moving cylinder; oscillations were removed by enforcing the local extrema within the interpolation stencil.

An improved approach is to use basis splines (better known as B-splines), which are piecewise polynomial functions with a local basis of minimal support for a given interpolation order and smoothness. However, there is no straightforward extension of splines from one dimension to two or more dimensions as the dimension of the spline space is unknown. For two or more dimensions, a set of source points can always be found that renders the interpolation problem singular [66]. Polynomial interpolation of all data is therefore not possible [47] and successive one-dimensional interpolations must be performed.

In spite of these limitations, polynomial interpolation is readily utilized because its simplicity facilitates implementation as well as error and stability analyses. As a result, much insight into the nature and applicability of the interpolation may be gained. Interpolation schemes may be explicit or implicit; in an *explicit* approach, interpolated solutions are calculated for receiver (or target) points that are decoupled from each other. This is not the case for overset grid configurations with insufficient overlap in which fringe points may be required to simultaneously donate and receive a solution during the data transfer. In this situation, the solutions for fringe points are coupled across multiple grids and an interpolated solution is not possible with an explicit approach. Alternatively, the coupled solution may be obtained with an *implicit* approach by solving a system of equations. This may be written in general form as [18]:

$$\sum_{k=1}^{BW} L_k \hat{f}(\hat{x}_k) = \sum_{j=0}^{N-1} R_j f(x_j). \tag{15}$$

L_k and R_j are interpolation coefficients, $\hat{x}_k$ are the target points, and $\hat{f}$ is the unknown function being represented. When the BW parameter is greater than one, the interpolation is implicit and a linear system of equations is formed. Furthermore, *compact* algorithms can be applied in the implicit approach that provide higher-order accuracy with specified spectral characteristics in the interpolated solution; this is accomplished by deriving new sets of interpolation coefficients [67]. A compact approach can offer spectral accuracy while providing geometric modeling flexibility.

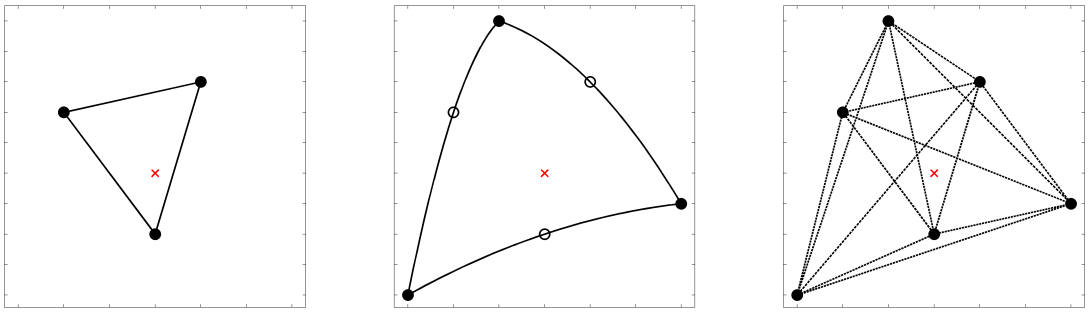
Sherer and Scott [18] developed an optimized compact interpolation method based on generalized Lagrange polynomials. Optimal interpolation coefficients were calculated to minimize the integrated error over a specified wavespace. This approach was demonstrated to theoretically reduce error by an additional order of magnitude over explicit methods by a one-dimensional Fourier analysis, but such improvements were not observed in practice. Instead, Sherer and Scott found that the global error was minimized at the expense of increased, non-monotonic error at lower wavenumbers [18]. The authors also noted diminishing returns for higher order polynomials (e.g., sixth-order vs. fourth-order).

An implicit approach is able to handle the solution of coupled fringe points and therefore can be applied to the problem of interpolation at orphan points. However, implementation on non-uniform meshes is not straightforward since a well-behaved, solvable system of coupled equations is not guaranteed for arbitrarily distributed points [18]. Therefore, applicability to unstructured methodologies and overset applications is limited since donor points are generally non-uniformly distributed. In addition, B-splines were found to be robust but more difficult to implement, requiring increased stencil sizes due to poor interpolation quality near stencil edges [18]. Finally, recent work on high-accuracy polynomial algorithms by Chicheportiche and Gloerfelt [68] has also demonstrated that interpolation stencils need to be tailored for a given level of accuracy.

3.1.2 Trilinear Interpolation

Trilinear interpolation (or bilinear interpolation in two dimensions) is the most common and efficient application of interpolating polynomials. The approach performs a sequence of one-dimensional linear interpolations, with the basis being the two-point Lagrangian polynomial [45]. This piecewise linear interpolation is a second-order approach (demonstrated by a simple Taylor series expansion) that produces a continuous function with discontinuous gradients at the given nodes. However, this approach is not always the fastest or the most accurate [69]. For overset schemes, Freitas and Runnels [32] have noted that traditional data transfer (e.g., with bilinear interpolation) between grids will introduce discontinuous mass fluxes and consequently mass conservation error. These may be mitigated by iterating until both the global solution is converged and the sum of mass fluxes is zero, and motivates the use of more accurate interpolation methods that can both better preserve fluid physics and accelerate convergence.

Additional layers of fringe points can be used to fit a tricubic (or bicubic in two dimensions) basis and yield a higher-order solution in terms of smoothness. Alternatively, f_i and its derivatives $\left(\frac{\partial f}{\partial x_1}\right)_i$, $\left(\frac{\partial f}{\partial x_2}\right)_i$, and $\left(\frac{\partial^2 f}{\partial x_1 \partial x_2}\right)_i$ can be specified and enforced at the boundary of the interpolation stencil. These derivatives may be calculated using numerical differencing or one-dimensional splines [46]. While this approach produces a smooth function, additional uncertainty is introduced as the quality of interpolation depends significantly on the specified derivatives which have inherent errors [14]. Despite these nuances, sophisticated variations on linear interpolation are widely used in overset chimera schemes, and a review of recent efforts is presented by Tang et al. [28]. In general, bilinear, trilinear, and biquadratic approaches all yield non-conservative pressure and velocity fields, which can negatively impact solution quality. Hubbard and Chen [70] found that trilinear interpolation at grid interfaces for all flow variables resulted in oscillatory pressure and velocity fields



(a) Three-node element (b) Six-node element (c) Scattered data

Figure 7: Examples of interpolation point clouds for different mapping methods associated with a target point (red marker).

in an incompressible viscous flow simulation.

Direct application of trilinear interpolation requires a structured grid with cell edges in alignment with the x, y, z axes. For body-fitted or unstructured grids, localized interpolation within arbitrarily shaped and oriented grid cells requires in general a mapping from the global space to a computational space, for both linear and higher-order interpolations. This mapping approach is described in the following section.

3.1.3 Inverse Isoparametric Mapping

Many modern CFD codes utilize body-fitted grids, which require that the mapping be applicable to a variety of computational cell geometries. The inverse isoparametric mapping (IIM) is a general finite element analysis technique applicable to all mesh types. The motivation behind the isoparametric formulation is to form a relationship between displacements at any point within an element and displacements at element nodes. These displacements are related through the use of interpolation or shape functions [71]. The approach is coined *isoparametric* because the same shape functions are employed to describe both the problem geometry and the problem unknowns

(e.g., displacements) with unknowns being scalar or vector quantities. The formulation eliminates the need to construct a transformation matrix between global and local ("natural") coordinate systems. General elements in three dimensions may be constructed with a variable number of interior nodes, thus constituting a cloud-based approach; examples are illustrated in Fig. 7a and b. Shape functions may be iteratively refined by increasing the number of nodes, with each additional interior node providing a correction to the existing shape functions [71]. In general, the highest accuracy is obtained by rectangular (or rectangular in a local plane for three dimensions) elements with intermediate nodes located at natural coordinate positions, the midpoints of edges [71].

An isoparametric mapping is one-to-one from coordinates relative to an interpolation stencil (in computational space) to a point in the global, physical space. Thus the *inverse* transformation (or IIM) is needed to uniquely specify the coordinates of a point in Cartesian computational space based on its location in physical space. These coordinates are also referred in the literature as *trilinear mapping coordinates* [16], *element parametric coordinates* [72], or *natural coordinates* [73]. Once the mapping coordinates ξ, η and ζ in each direction are found from the geometry and selected shape functions, the interpolation weights are known. The isoparametric mapping and interpolant is then:

$$\boldsymbol{x} = \sum_{i}^{n} W_i(\xi, \eta, \zeta)\boldsymbol{x}_i, \tag{16}$$

where W_i are the shape functions and interpolation weights at each node i.

In practice, elements are commonly defined as first-order [74] if their shape functions are first-order polynomials. In two dimensions, this corresponds to a four-node quadrilateral element or a three-node triangle element. These elements provide a single degree of freedom (e.g., displacements) at each node. A higher-order polynomial may be specified, permitting additional degrees of freedom (e.g., displacements and rotations) at each node. For instance, a quadratic triangular element is formed

using six nodes and is described by a quadratic shape function along each side of the element.

The mapping coordinates are determined by Eqn. 16, along with the condition that $\sum_i^n W_i(\xi, \eta, \zeta) = 1$ for completeness. For example, an IIM for a four-node quadrilateral (illustrated in Fig. 8) is determined by solving the following system:

$$\begin{bmatrix} 1 \\ x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{bmatrix}, \tag{17}$$

where

$$W_1 = (1-\xi)(1-\eta), \qquad W_2 = \xi(1-\eta), \qquad W_3 = \xi\eta, \qquad W_4 = (1-\xi)\eta. \tag{18}$$

This nonlinear system of equations can be written as $F([\xi\ \eta]^T) = 0$ for known global coordinates x, y, x_i, and y_i, and is readily solved by a Newton-Raphson technique. This approach is applied by a number of state-of-the-art connectivity codes to determine trilinear mapping coordinates and interpolation weights [72, 11, 10]. Alternatively, a bisection method may be applied to reduce computational cost for larger systems [75, 76]. Once the mapping coordinates (ξ,η) are known, substitution into Eqn. 18 provides the interpolation weights. The shape functions considered here (W_i) are defined so that the element is mapped to coordinates in $[0, 1]$, differing from conventional finite element definitions wherein the mapping coordinates range from $[-1, 1]$. Substituting the coordinates of each node in the natural coordinate system (Fig. 8) into Eqn. 18 results in weights of one at that node and zero at all other nodes, therefore satisfying Eqn. 16. In three dimensions, shape functions are defined in Ref. 16 for tetrahedra

$$W_1 = 1 - \xi - \eta\zeta, \qquad W_2 = \xi, \qquad W_3 = \eta, \qquad W_4 = \zeta \tag{19}$$

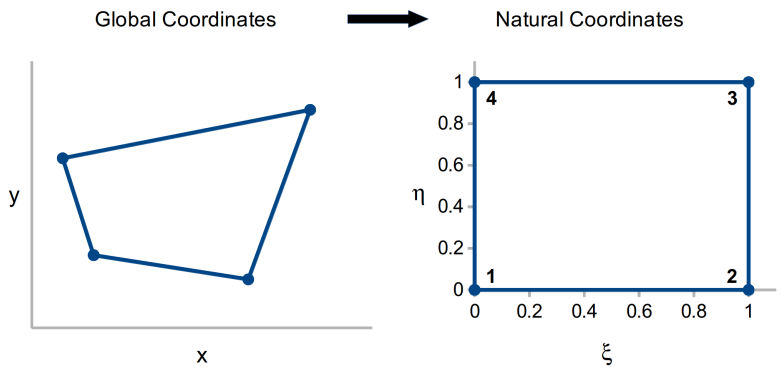


Figure 8: Inverse isoparametric mapping for a four-node quadrilateral computational element.

and hexahedra

$$
\begin{aligned}
W_1 &= (1-\zeta)(1-\xi-\eta+\xi\eta), & W_5 &= \zeta(1-\xi-\eta+\xi\eta),\\
W_2 &= (1-\zeta)\xi(1-\eta), & W_6 &= \zeta\xi(1-\eta),\\
W_3 &= (1-\zeta)\eta(1-\xi), & W_7 &= \zeta\eta(1-\xi),\\
W_4 &= (1-\zeta)\xi\eta, & W_8 &= \zeta\xi\eta.
\end{aligned}
\tag{20}
$$

Special Cases

In certain situations, an isoparametric mapping can be directly related to bilinear (or trilinear) interpolation. With a Cartesian grid topology, an IIM of four-node quadrilaterals (or eight-node hexahedra) can be directly calculated because the global coordinate system is similar to the natural coordinate system. Mapping coordinates (ξ,η,ζ) satisfying Eqn. 17 are directly evaluated found by scaling and translating the local cell to a unit square (or cube in three dimensions). Therefore the nonlinear solution procedure may be avoided altogether. For example, on a two-dimensional Cartesian grid with uniform spacing Δs, the i-th cell in the x direction and the j-th cell in the y direction are defined by the following nodes:

$$x_{i,j} = i\Delta s \qquad x_{i+1,j} = (i+1)\Delta s \qquad y_{i,j} = j\Delta s \qquad x_{i,j+1} = (j+1)\Delta s.$$

The mapping coordinates for an arbitrary point (x, y) are therefore:

$$\xi = \frac{x - x_{i,j}}{x_{i+1,j} - x_{i,j}} = \frac{x - i\Delta s}{\Delta s}$$
$$\eta = \frac{y - y_{i,j}}{y_{i,j+1} - y_{i,j}} = \frac{y - j\Delta s}{\Delta s}$$

.

For unstructured grids consisting of triangular (or tetrahedral) elements, classical linear interpolation [24] is accomplished by transforming Cartesian nodal coordinates to barycentric coordinates. In this system, any interior point is represented as the center of mass calculated from masses located at each vertex. At each vertex of the element, the barycentric coordinate or mass is unity. The sum of all coordinates must be one, and all non-negative values indicate that a point is on the interior of the element. This coordinate transformation is equivalent to an IIM with three-node triangles or four-node tetrahedral elements. The mapping for a three-node triangle element (illustrated in Fig. 9) has a form identical to Eqn. 17:

$$\begin{bmatrix} 1 \\ x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix}, \tag{21}$$

where

$$W_1 = 1 - \xi - \eta, \qquad W_2 = \xi \qquad W_3 = \eta. \tag{22}$$

This system of linear equations may be directly solved without an iterative procedure, and the weights W_i are the barycentric coordinates satisfying the isoparametric mapping (Eqn. 16).

Limitations

Performance of the IIM depends heavily upon the finite element. For instance, three-node triangle elements are better suited for approximating flat surfaces while six-node triangles are more suitable for highly-curved surfaces [73]. When interpolating

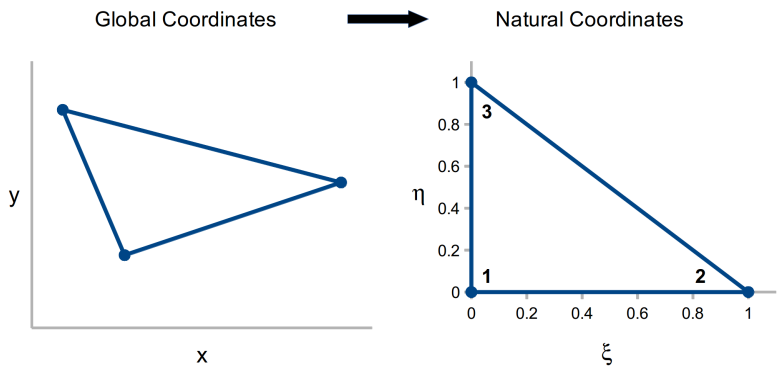


Figure 9: Inverse isoparametric mapping for a three-node triangular computational element.

conserved aerodynamic variables rather than displacements, flat and curved surfaces correspond to solution fields of varying smoothness. While the IIM approach affords additional modeling flexibility, a kernel interpolation approach (e.g., with radial basis functions discussed in Section 3.1.5) has the potential to be more robust and care-free. The IIM approach is not suitable when the target node is not within any element of the mesh since it is only applicable to interpolation [77, 73]. Application to non-contiguous interfaces requires projection of points outside the mesh onto the boundary [73]. Since extrapolation cannot be performed, the treatment of overset orphan points with IIM is not straightforward. To enable interpolation, additional non-fringe donor points will have to be sourced from more than one grid in order to form a cloud of donors that enclose the orphan point.

Typical applications of finite elements do not account for highly distorted or convex elements, which may occur when considering arbitrary distributions of points. Figure 10 illustrates the same set of five nodes and three possible element connectivities between these points. If higher-order finite elements are to be applied for interpolation, the elements will not necessarily be characterized by polynomial completeness. For instance, an eight-node quadratic "serendipity" element (i.e., a rectangular element with no interior nodes) can exactly represent a field described by a quadratic polynomial only when the element is rectangular; a nine-node Lagrangian element is

needed if the element is not rectangular [74]. For the most reliable analysis, finite elements should have straight sides and evenly-spaced nodes [78].

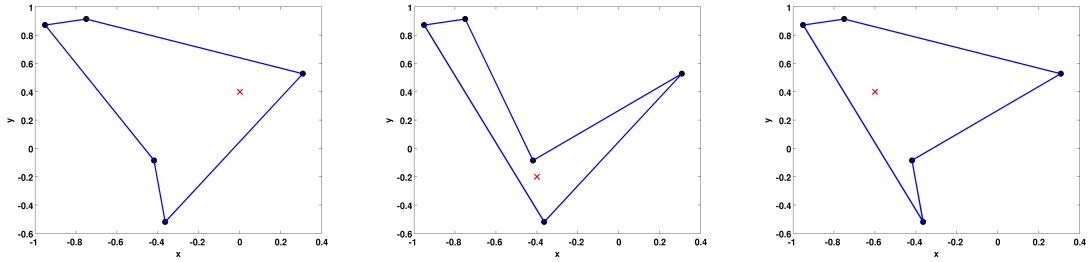


Figure 10: Sample element connectivities for an arbitrarily distributed set of donor points (filled black circles) and receptors (red marker).

In addition to the issue of completeness, a distorted element can also produce non-unique mappings due to the presence of concave vertices. By definition, a vertex is concave when its interior angle is greater than 180°. As an illustrative numerical example, a four-node quadrilateral element is considered. Initially the four-nodes form a rectangle but the lower-left corner (node 1) is shifted toward the upper-right corner (node 3) so that the interior angle varies from convex to concave (Fig. 11). The mapping coordinates of node 1 are determined with a Newton-Raphson technique, and the shape functions associated with each node is evaluated with these coordinates. Clearly, the shape function associated with node 1 should be unity when evaluated at node 1, and the shape function at all other nodes should be zero. However, this is not the case when the interior angle exceeds 180° (Fig. 11) as another set of mapping coordinates exists that also satisfies Eqn. 17 and maps to the global coordinates of node 1. Therefore the recovery of discrete input values is not guaranteed at element nodes. These issues may be mitigated by partitioning the element into multiple elements and deriving new shape functions [79] or applying a hybrid approach involving analytically derived stress functions [80].

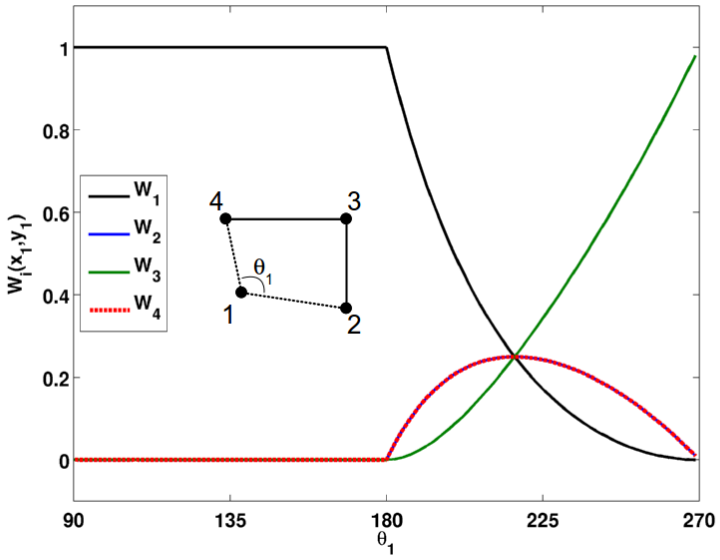


Figure 11: Shape function values for a four-node quadrilateral element evaluated at node 1. The W_2 and W_4 curves are coincident.

3.1.4 Inverse Distance Weighting

In comparison to stencil-based (polynomial) methods, neighborhood-based data transfer methods applying radial functions reduce a d-dimensional function (with $d > 1$) into a univariate function of Euclidean distance. These approach are therefore directly applicable to any dimension. A strength of using a neighborhood method for overset CFD is that connectivity information is not explicitly needed. As with kernel function interpolation techniques, how nodes are connected within each grid is not important as only their distances to points of interest are relevant to the scattered data transfer problem.

Inverse distance weighting (IDW) is one such technique, originally developed by Shepard [81] and later extended by Renka [82] to support trivariate interpolation. IDW is commonly applied in response-surface modeling applications (e.g., kriging or support vector machines) [83]. The method solves a moving least squares problem with radial weighting functions similar to the basis functions described in Section 3.1.5. In contrast to radial basis function (RBF) interpolation, this approach

is based on regression rather than interpolation. The difference is that a regression produces an *approximation* to the sampled data and unlike interpolation, does not guarantee that the approximate value at sample locations will be identical to the corresponding sampled values. Even so, it has been proven that the IDW approach is able to provide local polynomial reproduction properties which facilitate error analysis [47].

The basic form of the problem to be solved is [47]:

$$\min \left\{ \sum_{i \in I(x)} \left[f(x) - f(x_i) \right]^2 w(x - x_i) \right\} \tag{23}$$

This yields an approximation $s(x_j) \approx f_j$, using a set of points I within some neighborhood of x, where w is a compact translation-invariant weighting function. For reference, a recent successful application of this technique depended on a cloud of 10 to 15 of the nearest data points[84]. The weighting functions Shepard applied were [85]:

$$w(x) = \frac{1}{||x||_2^2} = \frac{1}{r^2}, \qquad w(x) = \frac{\exp(-\alpha r^2)}{r^2}$$

with $r \equiv ||x||_2$ being the radial distance. The final approximant is:

$$s_{f,X} = \sum_{j \in I(x)} f(x_j) \frac{w_j(x - x_j)}{\sum_{i \in I(x)} w_i(x - x_i)} \tag{24}$$

According to Wood [86], IDW is most effective at reproducing equally spaced sample points but may be ineffective at reproducing multi-scale data since it does not account for the spatial structure of the data [59]. While this approach has historical significance, the baseline IDW method has not been investigated since the moving least-squares problem is more expensive and can be less accurate than other neighborhood-based interpolation techniques (e.g., with radial basis functions, discussed in Section 3.1.5) near data sampling locations. A similar approach known as a *kernel regression* will be revisited in Chapter 4 as it formed the basis for an adaptive steering approach developed by other researchers [87].

3.1.5 Radial Basis Function Interpolation

Radial basis function (RBF) interpolation is a scattered data interpolation technique that is a specific type of kernel-based interpolation [88, 57]. Broadly speaking, *kernel function interpolation* uses a linear combination of basis functions (kernels) to represent highly nonlinear functions. The success of the approach lies in the selection of an appropriate kernel (a nonlinear basis) and source points for the interpolation. Determination of interpolation weights is straightforward and discussed in Section 3.1.5. On a historical note, the concept of a "kernel" dates back to the original Gaussian heat kernel [89], which depending on context, can be a fundamental PDE solution (a basis function), a symmetric operator, or a distribution function representing a stochastic process. In the context of this thesis, a kernel is conceptually equivalent to a d-dimensional multivariate function, e.g., $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$.

Among the plethora of kernel functions used for interpolation, the most popular by far are the radial (or *isotropic*) kernels that give RBF interpolation its name. Some common RBFs are presented in Table 1. These functions have the property that they are translation-invariant, i.e., they are independent of Euclidean transformation [90]. While their *centers* can change, the functions themselves are everywhere only a function of the Euclidean distance from the center. This definition does not include affine transformations that include scaling operations that can deform the circular kernel footprint (into say, an ellipse) which would result in a non-traditional RBF interpolation approach. *Anisotropic* kernel function interpolations are discussed in Chapter 4.

Recommendations from a number of authors [85, 75, 91, 77, 47] have suggested that RBF interpolation is a general, accurate approach to data transfer. Scattered data methods are especially attractive for solution transfer because:

1. They permit interpolation (and extrapolation) based on arbitrarily clustered clouds of points in any dimensional space.

Table 1: Sample global radial basis functions, $\phi(r)$ with $r = ||x||_2$.

Name	Definition	Parameter	Positive Definite
Gaussian	$e^{-\alpha r^2}$	$\alpha > 0$	yes
Polyharmonic Splines	r^k	$k = 1, 3, 5...$	conditional
	$r^k \log r$	$k = 2, 4, 6...$	conditional
Thin-Plate Spline (TPS)	$r^2 \log r$	n/a	conditional
Multiquadrics (MQ)	$\sqrt{r^2 + c^2}$	$c > 0$	conditional
Inverse Multiquadrics	$1/\sqrt{r^2 + c^2}$	$c > 0$	yes

2. They have in general higher-order accuracy that can be increased by freely adding data points.

3. They are directly applicable to unstructured methodologies since the interpolant is decoupled from the computational mesh so that there is no requirement on the spatial structure of the sampled data.

4. They are readily applied to the problem of solution transfer in overset methods since they do not require connectivity information.

Positive Definiteness

A basis function or kernel (K) is considered positive definite if its kernel matrix $K_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ (which has an equivalent form as the interpolation matrix given in Eqn. 29) satisfies:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j K_{ij} > 0 \tag{25}$$

for distinct points $\boldsymbol{x}$, and any vector of non-zero values $\boldsymbol{c}$ (e.g., the interpolation coefficients). The positive definiteness of a kernel is dimension-dependent [66] and is essential to the problem of scattered data approximation because it guarantees the existence of an invertible interpolation matrix [47]. It is known that global RBFs are either positive-definite or conditionally positive-definite, while compactly-supported functions (discussed in Section 3.1.5) are guaranteed to be positive-definite [90]. In

the conditional case, a polynomial must be added to the interpolant (Eqn. 26) for solvability and uniqueness [66] but this is optional in the positive-definite case. To improve the reproduction quality of approximately constant or linear fields, the polynomial terms have been included for all RBF interpolation cases in the present work.

In general, positive-definite kernels may be arbitrarily derived to satisfy dimensionality and smoothness constraints. Using Fourier transforms, Bochner [92] showed that positive semi-definite radial functions can be characterized as monotonic, and led to the conclusion that positive-definite functions are essentially equivalent to completely monotone functions [57]. This framework has since been used to recursively derive a number of compactly-supported basis functions by convolution [47, 93], including those shown in Table 2.

Solution Procedure

Table 1 presents some well-known radial basis functions, defined in general as $\Phi(\boldsymbol{x}) = \phi(r), r = ||\boldsymbol{x}||_2$. Some of these functions have behaviors that depend on shape parameters that are not necessarily constant throughout a solution domain.

RBF interpolants have the following form:

$$s_{f,X}(x) = \sum_{j=1}^{N} \alpha_j \Phi(x - x_j) + \sum_{k=1}^{Q} \beta_k p_k(x), \tag{26}$$

where s is the RBF interpolant evaluated at an arbitrary location x; α_j are the interpolation coefficients; and x_j are the RBF *centers* chosen to be coincident with the set of sampled *data sites* (or source points), X. The quality of the results is sensitive to kernel centers and widths [88], and in the case of the centers coinciding with the locations at which the solution is known (i.e., the data sites), an interpolant exists per the Mairhuber-Curtis theorem [47]. p_k is a set of linearly independent functions $p_1, p_2, ..., p_Q$ on $\mathbb{R}^d$ (with Q being the dimension of the function space), and ensures a unique solution and solvability of the interpolation matrix by making the system conditionally positive definite [47, 90]. Typically, p_k is chosen to be a

polynomial basis. An additional constraint is placed on the function p_k:

$$\sum_{j=1}^{N} \alpha_j p_k(x_j) = 0. \tag{27}$$

This constraint makes the interpolation coefficients orthogonal to the polynomial space by removing extra degrees of freedom from the polynomial [94] and controlling the rate of growth at infinity of the non-polynomial term [95].

Eqns. 26 and 27 are combined and solved as a linear system of equations:

$$\begin{bmatrix} A_{\Phi,X} & P \\ P^T & 0 \end{bmatrix} \begin{Bmatrix} \alpha \\ \beta \end{Bmatrix} = \begin{Bmatrix} f|_X \\ 0 \end{Bmatrix}. \tag{28}$$

The submatrices are defined as:

$$A_{\Phi,X} = (\Phi(x_j - x_k)) \in \mathbb{R}^{N \times N} \tag{29}$$

$$P = (p_k(x_j)) \in \mathbb{R}^{N \times Q}. \tag{30}$$

Numerous techniques exist for solving the system in Eqn. 28. For smaller interpolation problems, directly inverting the left-hand side matrix is effective. However, for typical two- and three-dimensional problems, the interpolation matrix quickly becomes intractable for a global (rather than neighborhood-based) approach that incorporates the entire set of available data points.

Implementation Challenges

The potential challenges in applying RBFs are widely understood [96, 97]. Effective interpolation requires a trade-off between the number of source points (generally increasing accuracy with number) and numerical stability (which decreases with number due to ill-conditioning of the interpolation matrix). Even if the matrix inversion can be reliably performed, the cost of a direct evaluation (using Gaussian elimination) is $\mathcal{O}(N^3)$ [98], where N is the number of source points for the interpolation. If an indirect, iterative approach is used, the computational cost will be comparable to the

cost of solving the governing physical equations at the same points. Thus the interpolation cost can quickly become a non-negligible portion of the overall simulation cost. Even if cost were not prohibitive, application of global basis functions can result in spurious fluctuations [73]. Solution degradation near clustered grid points is also of potential concern [73], motivating a localized RBF approach.

A practical engineering solution is to employ a smaller subset of points (i.e., a cloud of points) within a neighborhood of the target point for each data transfer. If the reconstruction of an entire solution field is required, a domain decomposition or *partition of unity* approach should be used to guarantee error bounds and convergence properties when using global basis functions [47]. This approach separates a complete global domain into subdomains on which smaller interpolation problems are solved. A global solution is formed using a weighted sum of the local solutions, where the weighting comes from the partition of unity. If only local solutions are needed, then the use of compactly-supported basis functions will be numerically stable and more physically consistent (Section 3.1.5).

Many applications of RBF data fitting involve smooth, continuous solution fields. In practice, sampled data (e.g., from experiment and/or turbulent flow fields) will not be free from noise, and RBF approaches may overfit the input data [83]. The problem can be mitigated by through regularization, which adds a shift along the diagonal of the interpolation matrix, i.e., $A \rightarrow A + \lambda I$, where I is the identity matrix and λ is the regularization parameter typically set to the variance of the noise. Another alternative is to use fewer bases than sample locations, thereby recasting the interpolation problem as a least-squares error minimization problem. In both cases, additional uncertainty is introduced into the problem, and therefore these approaches have not been considered in the present work.

Table 2: Sample compactly-supported radial basis functions, $\phi(r)$ with $r = ||x||_2$ and $\phi(r > 1) = 0$.

Name	Definition
Wendland C^0 [47]	$(1-r)_+^2$
Wendland C^2 [47]	$(1-r)_+^4(4r+1)$
Wendland C^4 [47]	$(1-r)_+^6(35r^2+18r+3)$
Buhmann Thin-Plate Spline [93]	$2r^4 \log r - \frac{7}{2}r^4 + \frac{16}{3}r^3 - 2r^2 + \frac{1}{6}$

Compactly-Supported Basis Functions

A straightforward and mathematically rigorous approach to localized RBF interpolation is to use compactly-supported basis functions, an approach that was originally proposed by Schaback [90]. In contrast to the globally supported basis functions listed in Table 1, compactly-supported basis functions are by definition zero above a nominal radial distance equal to one. Support intervals other than $[0,\ 1]$ may be obtained through scaling. With these basis functions, sparse interpolation matrices (Eqn. 29) can be formed through judicious specification of a support radius based on the selected source points. Sparse matrices have the benefit of reduced memory requirements and reduced operation count when solving the interpolation system. An additional benefit of compactly-supported functions is that they are typically defined by functions that lend themselves to rapid evaluation [47]. Some sample compactly-supported RBFs are listed in Table 2. For the Wendland functions, the leading term is defined as

$$(1-r)_+ = \begin{cases} 1-r, & r \leq 1 \\ 0, & r > 1. \end{cases} \tag{31}$$

All Wendland functions listed have been derived with guaranteed positive definiteness in up to three dimensions. These functions have been derived with a prescribed smoothness C^k where k is the number of times the function may be differentiated and still preserve a zero slope at $r = 1$.

The condition of the interpolation matrix (Eqn. 29) is directly related to the numerical stability of solving the interpolation system (Eqn. 28) and tends to be relatively poor when using classical global basis functions (Table 1). With the exception of the Gaussian and Inverse Multiquadrics functions, all the other basis functions behave as a positive power of r. The interpolation matrix therefore has relatively large terms on the off-diagonal and a correspondingly large condition number. The Gaussian and Inverse Multiquadrics functions decrease with increasing r but never reach zero. In contrast, the compactly-supported functions are by definition zero for $r > 1$ and can provide sparse interpolation matrices with the largest values on the diagonal. As a result, the compactly-supported basis functions have a lower condition number [99] and is therefore more numerically stable. The inverse relationship between condition number and interpolation error is discussed in Ref. 100.

The remaining unknown when implementing a compactly-supported RBF interpolation is the selection of a support radius, i.e., the extent of the region where the basis function is non-zero. Larger supports in general yield improved quality results for the purposes of constructing (smooth) geometric surfaces [90] simply because more information about the solution field is utilized, and the limit of increasing support region size is the entire computational domain. When selecting a support size, the goal is to obtain comparable rather than more accurate results with respect to the global approach. As with polynomial interpolants, there is a caveat in regard to the behavior of the sample solution. For example, polynomial interpolation achieves high accuracy only when the function inside the interpolation stencil is smooth, otherwise the solution will become oscillatory [62].

3.1.6 Other Techniques

Non-Traditional RBFs

While the radial bases have been more widely studied, many other basis functions have the same attractive properties. One new basis function is the multiscale kernel

developed by Opfer [101]. This compact multiscale approach draws from wavelet theory and requires no user manipulation of the data in terms of rescaling. The approach replaces the radial basis with a superposition of shifted single-scale kernels:

$$K(x,y) = \sum_j \lambda_j \Phi_j(x,y) = \sum_j \lambda_j \sum_k \phi(2^j x - k)\phi(2^j y - k), \tag{32}$$

where λ_j are the interpolation coefficients, and k is an integer. As an added benefit, reconstructions of noisy data may be more accurate.

Another example is the Brownian bridge kernel, which is compact, positive-definite, and non-radial. It is typically used in modeling stochastic processes (e.g., in finance applications) [57]:

$$K(x,z) = \prod_{j=1}^{d} \min\{x_j, z_j\} - x_j z_j. \tag{33}$$

Its effectiveness in modeling fluctuating quantities makes it a candidate for interpolating turbulent flow field quantities.

Walsh Functions

Global solutions to nonlinear PDEs with discontinuities have been successfully derived using Walsh functions. Gnoffo [102] successfully modeled a shock in a quasi-one-dimensional nozzle where the Gibbs phenomenon was not observed. The effectiveness of the approach in representing nonlinearities lies in the self-similar mathematical property associated with Walsh functions, that is, the product of two Walsh functions is another Walsh function. This approach is applicable to problems involving data transfer but has not been evaluated for problems of interest due to computational complexity.

3.2 Selection of Mapping Techniques

In modern computational science, the standard practice for approximating data within a range of discrete points with a known solution is to apply linear interpolation, i.e.,

interpolation with a first-order polynomial. This approach is easy to implement, computationally efficient, and numerically stable in that no new local extrema are created. While this approach may be more effective in high-gradient regions because higher-order problems involving overshoot and undershoot are avoided, the scheme is also dissipative and can be inaccurate for nonlinear solution fields depending on the spacing of sampled data. The state of the art for data transfer between arbitrarily overlapping grids with potentially different grid topologies is to map global coordinates to local element coordinates to enable trilinear interpolation. This typically involves applying the IIM technique with linear basis functions, with mapping coordinates determined by Newton's method [72, 11, 10]. Though not explicitly described as an inverse isoparametric mapping, this is the approach employed by Suggar++ [16] and PUNDIT [10].

For increased accuracy, single-directional polynomials may be readily extended to higher-order. While this approach is simple and computationally inexpensive, the interpolated solution is susceptible to spurious oscillations (Figs. 5 and 6) and can require a filtering technique to minimize error. To address the issue of orphan points on overlapping grids, an implicit polynomial interpolation approach can be applied that permits fringe points to both donate and receive solutions. For general configurations, fringe points may come from one or more grids. However, implementation is problematic for uneven or arbitrary distributions of points since the system of equations is not guaranteed to be solvable.

The investigations in this thesis are centered around cloud-based rather than polynomial-based approaches to interpolate arbitrarily distributed data. Fundamentally, the difference between polynomial and scattered data techniques lies in the choice of basis function, and the properties of these functions dictate their applicability and implementation. A radial basis function is a nonlinear univariate function in which the indeterminate variable is a function of Euclidean distance, defined for

any dimension. The RBFs considered in the present work have interpolants that satisfy the mathematical property of local polynomial reproduction [66]. In comparison, the application of traditional polynomial interpolation in higher dimensions involves successive interpolations in each dimension [45]. In the form of Eqn. 28 (reproduced here),

$$\begin{bmatrix} A_{\Phi,X} & P \\ P^T & 0 \end{bmatrix} \begin{Bmatrix} \alpha \\ \beta \end{Bmatrix} = \begin{Bmatrix} f|_X \\ 0 \end{Bmatrix},$$

where the submatrix P is optional for radial basis function interpolation. Conversely, for polynomial interpolation, the radial basis function interpolation matrix A is eliminated and the problem is only solved for the β coefficients. When interpolating with polynomials in higher than one dimension, a set of source points that forms a singular interpolation problem can always be found [66].

While the use of cloud-based techniques may incur additional cost, the strength of this approach is that it is completely stencil-free. This lack of dependence on an interpolation stencil means that the interpolation does not require any connectivity information and points may be arbitrarily included for improved accuracy. Such an approach naturally precludes the existence of orphan points since additional donor points can always be sourced from the vicinity of the target to permit either interpolation or extrapolation. To minimize uncertainties for use with all types of solution fields, basis functions that depend on shape parameters have not been considered because improper selection of a shape parameter can result in an ill-conditioned matrix [91] and/or reduced accuracy. Recent work applying the Gaussian RBF to interpolating unstructured, three-dimensional particle velocimetry data yielded accurately interpolated flow fields with improved mass conservation [103], but least-squares optimization was required to determine shape parameters. Other general methods for selecting a shape parameter exist, including leave-one-out cross validation (LOOCV), generalized cross validation, and a maximum likelihood estimator [104]. All of these

methods incur additional cost; for example, a LOOCV algorithm requires numerous repeated evaluations of the interpolant with varying subsets of input data.

Focus is therefore on two basis functions, the thin-plate spline (TPS) recommended in Refs. 91 and 77 for accuracy and robustness in both interpolation and extrapolation, and the Wendland C^2 (W2) basis function recommended in Refs. 105, 106, and 107. Definitions of the thin-plate spline

$$\phi(r) = r^2 \log(r)$$

and compactly-supported Wendland C^2 function

$$\phi(r) = \begin{cases} (1-r)^4(4r+1), & r \leq 1 \\ 0, & r > 1 \end{cases}$$

are repeated here for reference. The derivation of the W2 function [47] ensures continuity up to the second derivative (C^2 smooth). Since typical CFD meshes have high levels of resolution, the increased smoothness of higher-order functions within the Wendland C^k family of functions (with $k > 2$) provides no added benefit [99] and therefore has not been considered. Rendall and Allen also observed that for proper selection of points, interpolation accuracy appears to be independent of flow, mesh type, and mesh density [99]. Prior work with these functions has focused on applicability to non-contiguous fluid-structure interfaces rather than applicability to general data type interpolations. Recent work in this area has applied the Wendland function only to interpolation across non-matching Cartesian grid interfaces rather than unstructured overset grids [107]. To demonstrate the effectiveness of RBF interpolation for general data transfer relative to current methods, state-of-the-art linear mappings (with triangle and quadrilateral elements) and a higher-order IIM with quadratic triangle elements have been evaluated.

3.3 Evaluation of IIM and RBF Techniques

Two studies have been performed on two-dimensional structured grids to compare the effectiveness of the mesh-free radial basis function (RBF) approaches to the established inverse isoparametric mapping (IIM) technique. Results and conclusions from the two-dimensional tests are assumed to be readily extensible to three dimensions. The IIM method is used for comparison because it has already employed by other researchers for unstructured methodologies, is general to both structured and unstructured meshes, and can be extensible to higher-order.

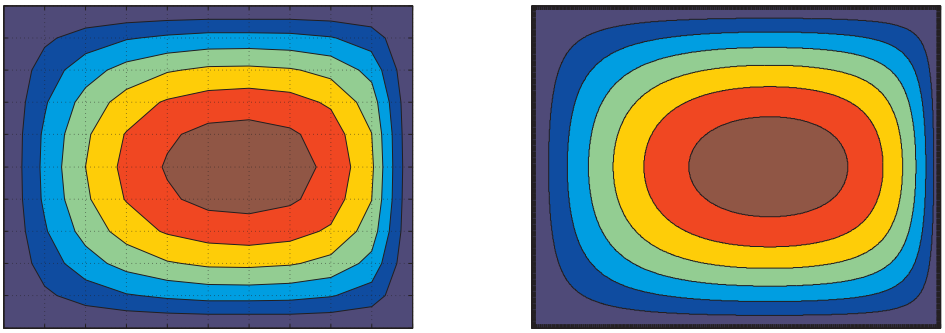
3.3.1 Elliptic PDE

An initial test case based on a PDE solution satisfying Dirchlet boundary conditions was studied on a unit square domain, $[0,1] \times [0,1]$. The specified closed-form solution is:

$$f(x,y) = 10xy(1-x)(1-y)\exp(x^{4.5}) \tag{34}$$

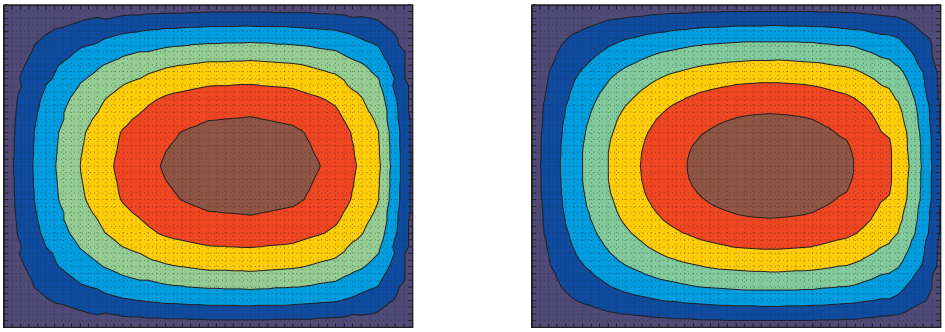
The interpolation target grid contained 41×41 points, with source grids ranging in size from 11×11 to 161×161. This encompasses both coarse-to-fine and fine-to-coarse interpolations. A grid spacing ratio has been defined as the ratio of output grid spacing to input grid spacing; values less than one indicate coarse-to-fine interpolation. To facilitate comparisons between structured and unstructured grid methods, the same Cartesian mesh was utilized for all cases. To form an unstructured mesh, quadrilateral cells were bisected into isosceles triangles. Higher-order triangular elements were formed by selecting a central triangle and appending all neighboring triangles that share a common edge.

The baseline interpolation scheme under investigation is the IIM method using four-node quadrilateral ("quad") elements, which is equivalent to bilinear interpolation on the Cartesian mesh and may be directly computed (discussed in Section 3.1.3, "Special Cases"). An analogous approach for unstructured grids involving three-node



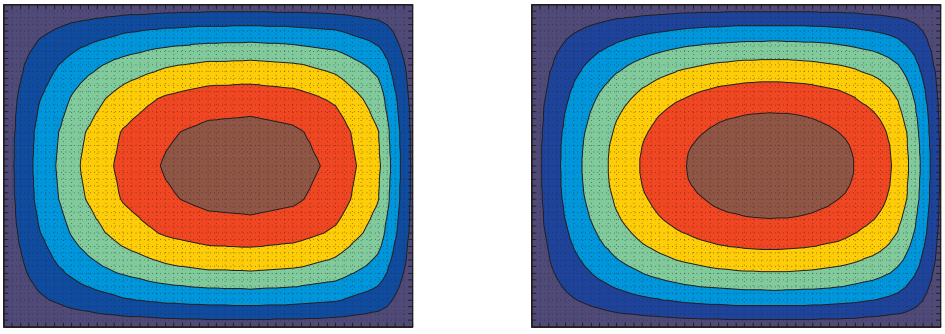
(a) Coarse interpolation source mesh (b) Exact solution

Figure 12: Discretized input (11×11) and exact solutions for analytical PDE.



(a) Linear three-node triangles (tri3) (b) Quadratic six-node triangles (tri6)

Figure 13: Inverse isoparametric mapping solution (from a coarse 11×11 mesh) for the analytical PDE.



(a) 4-node cloud (b) 16-node cloud

Figure 14: RBF interpolated solution (from a coarse 11×11 mesh) for the analytical PDE test case using the TPS basis, with local and neighborhood support.

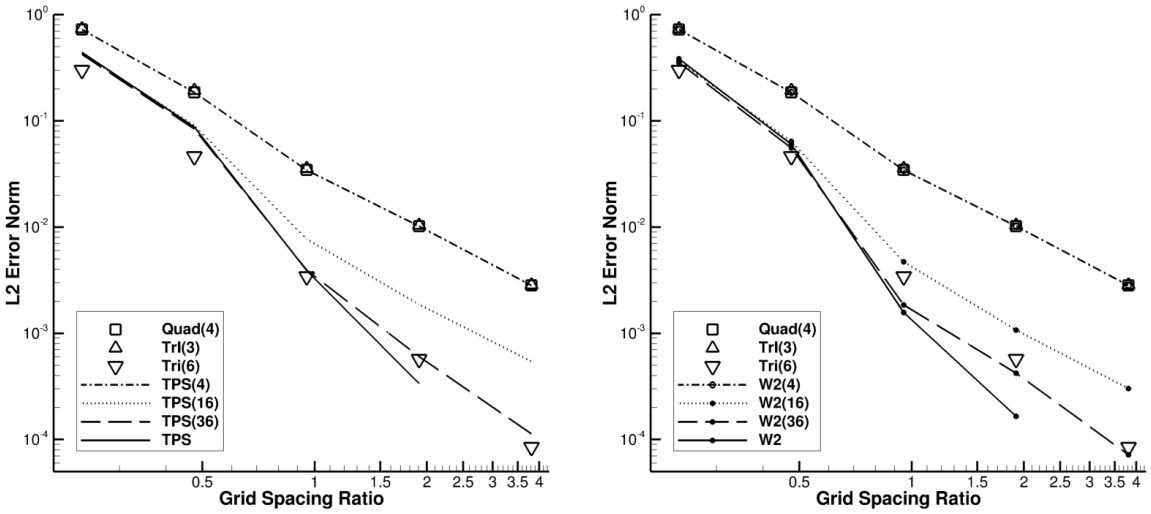
triangular ("tri") elements was considered for this test case, in addition to higher-order six-node triangular elements. Mapping coordinates for the three-node triangle were directly determined from barycentric coordinates (discussed in Section 3.1.3, "Special Cases") while the coordinates for the six-node triangle were calculated iteratively with Newton's method as in a general IIM application. These methods were compared with local, neighborhood, and globally supported schemes for RBF interpolation using thin-plate splines (TPS) and Wendland C^2 (W2) basis functions. Interpolation solutions for lower- and higher-order triangular element mappings (Fig. 13) and TPS RBF (Fig. 14) interpolation are depicted for the case with the greatest mesh disparity (mapping from a coarse 11×11 grid to a fine 41×41 grid). Three interpolation configurations with these functions have been considered with results detailed in the following subsections.

Local Interpolation

Only data from a single cell was included in the interpolation, mimicking the standard implementation in current domain connectivity software with donor points coming from a single computational cell. Local interpolation approaches included bilinear interpolation within a four-node quad element, a linear mapping within a three-node triangle element, and RBF interpolation with four source points. All interpolations including four nodes had virtually identical errors (to within 10^{-4}), plotted in Fig. 15. The three-node linear mapping had marginally higher errors (by 10^{-3} on average) than the four-node interpolation. Second-order accuracy was achieved for all methods, with a slope of approximately -1.8.

Neighborhood Interpolation

A cloud of source points was formed by including nodes from the enclosing cell in addition to nodes of surrounding cells. The number of donors in this case increases from four up to sixteen (in two dimensions). While it is straightforward to uniformly



(a) Thin-plate spline (b) Wendland C^2

Figure 15: L2-norm of the interpolation error at different grid spacing ratios for the analytical PDE solution. Legend names refer to the IIM element (quadrilateral or triangular) or the radial basis (thin-plate spline or Wendland C^2) used, and the number in parenthesis indicates the number of nodes included in the interpolation.

increase the number of points in all directions, it is not necessary. Gradient information about the solution field may be utilized to strategically add donor points and will be discussed further in Chapter 4. For coarse-to-fine interpolation, the neighborhood approach has up to 40% and 50% less error than the local approach when applying the TPS and W2 approaches, and 60% less error when applying the quadratic (six-node) triangular element. For fine-to-coarse interpolations, RBF errors were reduced by nearly an order of magnitude and six-node error was reduced over an order of magnitude. However, while the higher-order IIM has the lowest error of all the neighborhood approaches, Figure 13 reveals an interpolation artifact near the right edge of the domain where the solution is distorted in a relatively high gradient region. This suggests that the IIM approach may be more susceptible to dispersive errors when transferring data from coarse to fine grids. A similar artifact was not observed in the RBF solutions. In the coarse-to-fine range, errors for all neighborhood-based methods were approximately equal. In the fine-to-coarse asymptotic range, the W2 errors

were approximately 40% less than with the TPS basis. Both RBF approaches had second-order accuracy in the asymptotic range (with slopes of approximately -1.8), while the six-node IIM exhibited faster convergence (slope of -2.8).

If points are added to the cloud from an additional level of surrounding cells, then the total number of source points increases to 36. In this case, the observed error was comparable to the higher-order IIM results, with a comparable rate of convergence (summarized in Table 3). For fine-to-coarse interpolation (grid spacing ratio greater than one), the error for the IIM and RBF approaches are on the same order of magnitude, with TPS error being higher than IIM by 34% and W2 being lower by 15% at the finest grid spacing.

Table 3: Convergence rates for the analytical PDE solution when applying neighborhood and global interpolation approaches.

Method	Source Points	Slope
IIM six-node triangle	6	-2.8
Thin-plate spline	16	-1.8
	36	-2.4
	all	-3.5
Wendland C^2	16	-1.8
	36	-2.6
	all	-3.3

Global Interpolation

The limit of the neighborhood increasing in size is a cloud that includes all the points of the input mesh. This global approach is applicable to the RBF approach and permits simultaneous solution of the interpolated values at all target points. The resulting interpolated field was the most accurate when interpolating from fine to coarse grids with the W2 function, which had over 50% less error than the next best result obtained by the TPS basis, and up to two orders of magnitude less than the

linear approaches utilizing three and four node elements. In addition, the rate of convergence is higher than the six-node IIM, with slopes of -3.5 and -3.3 for the TPS and W2 basis functions respectively. This verifies that both accuracy and convergence rate increase with the number of source points with the RBF approach.

However, computational cost renders a global approach unusable in practical applications and numerical instability will decrease accuracy for larger problems. For grid spacing ratios greater than two (corresponding to an input mesh with 161×161 points, data were not obtained due to memory limitations in performing a direct solution of Eqn. 28. From Eqns. 29 and 30, the number of elements in the left-hand side is $(N+3)^2$ in two dimensions, with $N = 161 \times 161 = 25,921$. For double precision, the required storage for this matrix will be 5.4 gigabytes.

3.3.2 Sample Rotor Blade

This test case represents a more realistic fluid dynamics problem of interest and adds complexity through grid stretching and point clustering on a non-Cartesian structured domain. Repeated solution transfers of a surface pressure solution between fine and coarse grids (4100 and 600 nodes, respectively) were performed to study how the interpolated solutions behaved over multiple iterations corresponding to multiple interpolations. In this case, a single iteration consists of two solution transfers: from the fine grid to the coarse grid, and then from the coarse grid back onto the fine grid. After each iteration, the solution was compared with the original solution to quantify interpolation error. Interpolations with the IIM approach included three and six-node triangle elements, as well as four-node quadrilateral elements. These results are compared with the local and neighborhood cloud-based approaches described in the previous section, which applied thin-plate splines and Wendland C^2 interpolation. Since the grid is no longer Cartesian as in the PDE test case (Section 3.3.1), bilinear interpolation can no longer be directly performed with the four-node quadrilateral

element and Newton's method was applied to iteratively obtain mapping coordinates.

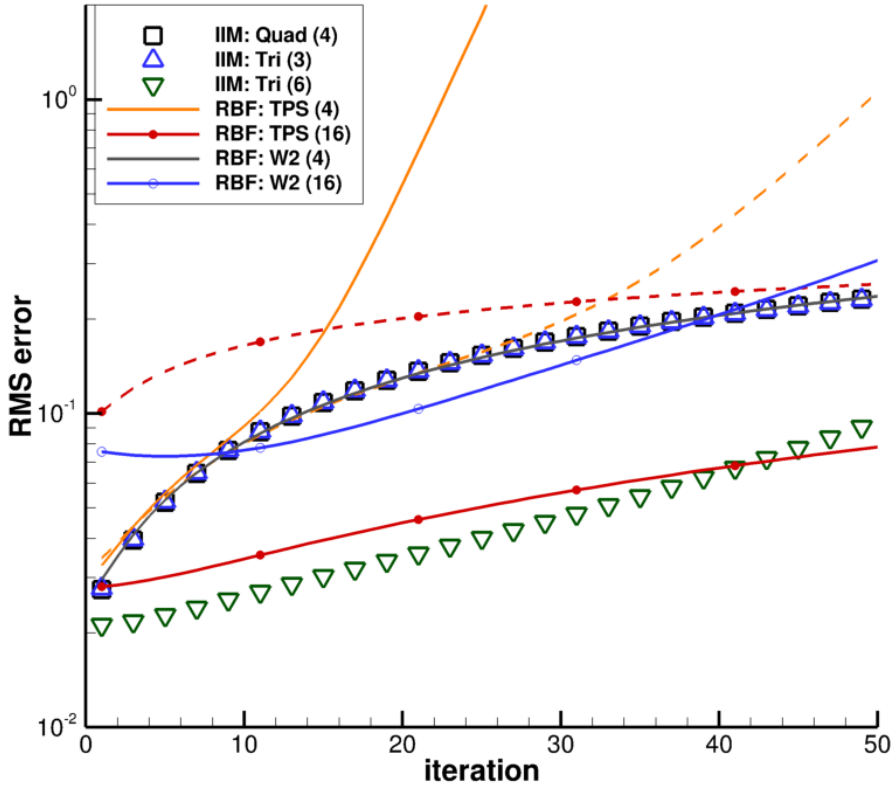


Figure 16: Root-mean-square error for repeated interpolations to and from fine and coarse meshes for the rotor blade test case. Solid lines denote cases where the neighborhood of interpolation points has been rescaled to a unit square while the dashed lines indicate that no normalization was applied.

The short-term evolution of root-mean-square (RMS) error (plotted in Fig. 16) has the following implications:

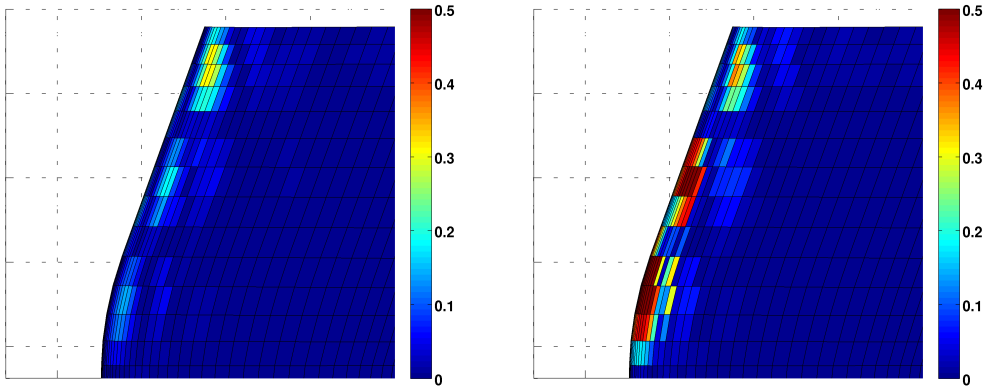
1. The growth in interpolation error when using a local interpolation (based on information from a single cell) is identical for the linear IIM approaches (four-node quadrilateral and three-node triangle elements) which agrees with the findings of the previous section. In other words, when a minimal amount of information is available, higher-order approaches can be no more effective and may be less stable (e.g., TPS interpolation supported by four nodes).

2. The higher-order IIM approach (six-node triangles) initially has the lowest error,

with nearly an order of magnitude reduction over the linear approaches at best. However, the error increases at a greater rate than the TPS (operating in a neighborhood of sixteen nodes). In comparison, the TPS produces nearly 70% less error than the linear approaches.

3. The effects of normalizing the input domain are apparent in comparing the un-normalized (dashed lines) and normalized (solid lines) error curves. If there are sufficient data to support the interpolation, locally rescaling the extents of the interpolation cloud to a unit domain improves accuracy by nearly an order of magnitude. However, when the input data is insufficient to maintain a stable solution, rescaling has a further destabilizing effect. For the four-node TPS approach, the errors diverged from other linear approaches after 10 iterations rather than 30 in the un-normalized case.

4. The W2 results tended to be less accurate than TPS in this case, due to the accumulation of errors in parts of the blade grid with high solution gradients and skewed, high aspect-ratio cells (Fig. 17).



(a) TPS basis, 16-node support (b) W2 basis, 16-node support

Figure 17: Rotor blade tip error after a single iteration of interpolations.

In the long-term (Fig. 18), all approaches were unstable in that the solution continued to drift from the initial solution with repeated data transfers. However,

the higher-order methods were the most unstable. Between the most accurate approaches (six-node triangle IIM and sixteen-node TPS), the IIM approach diverges more quickly. In comparison, the TPS approach is able to perform half an order of magnitude more iterations before errors surpass the linear level. Errors for the TPS methods with and without normalization approach each other and diverge at the same rate.

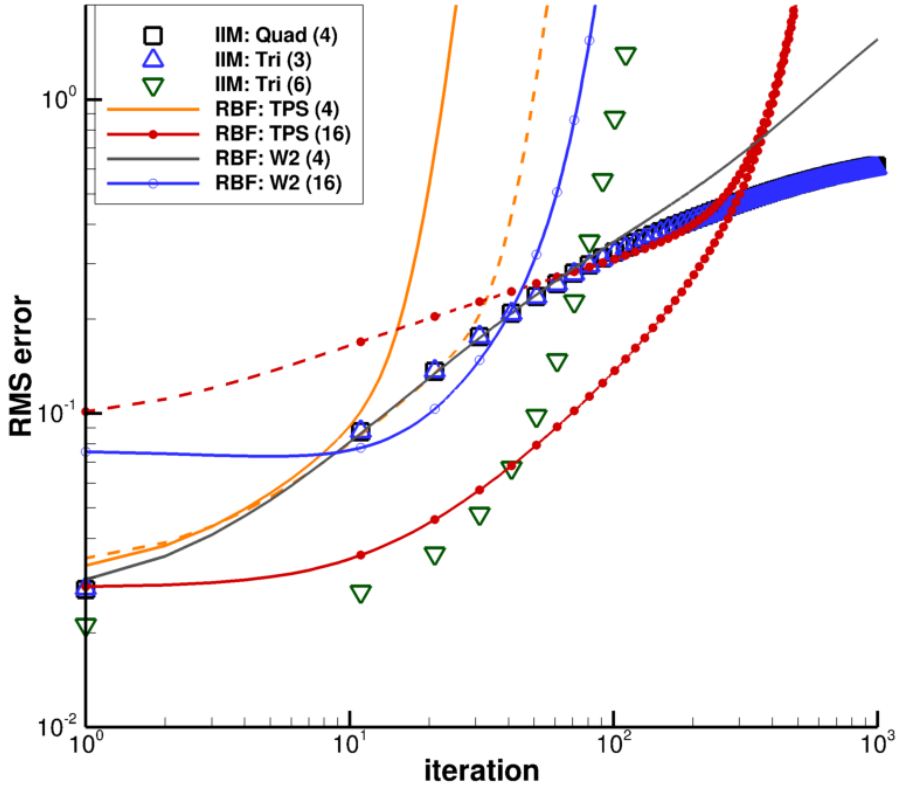


Figure 18: Root-mean-square error in the asymptotic range for repeated interpolations to and from fine and coarse meshes for the rotor blade test case. Solid lines denote cases where the neighborhood of interpolation points has been rescaled to a unit square while the dashed lines indicate that no normalization was applied.

A qualitative comparison of the different interpolation approaches is presented in Fig. 19. The baseline IIM approach with 4-node quads is dissipative and quickly smears the suction peak (blue-colored contours) over the entire blade surface. In contrast, a higher-order IIM approach with 6-node triangles maintains the blade surface solution despite some initial smearing (as quantified in Fig. 16). After 50 iterations,

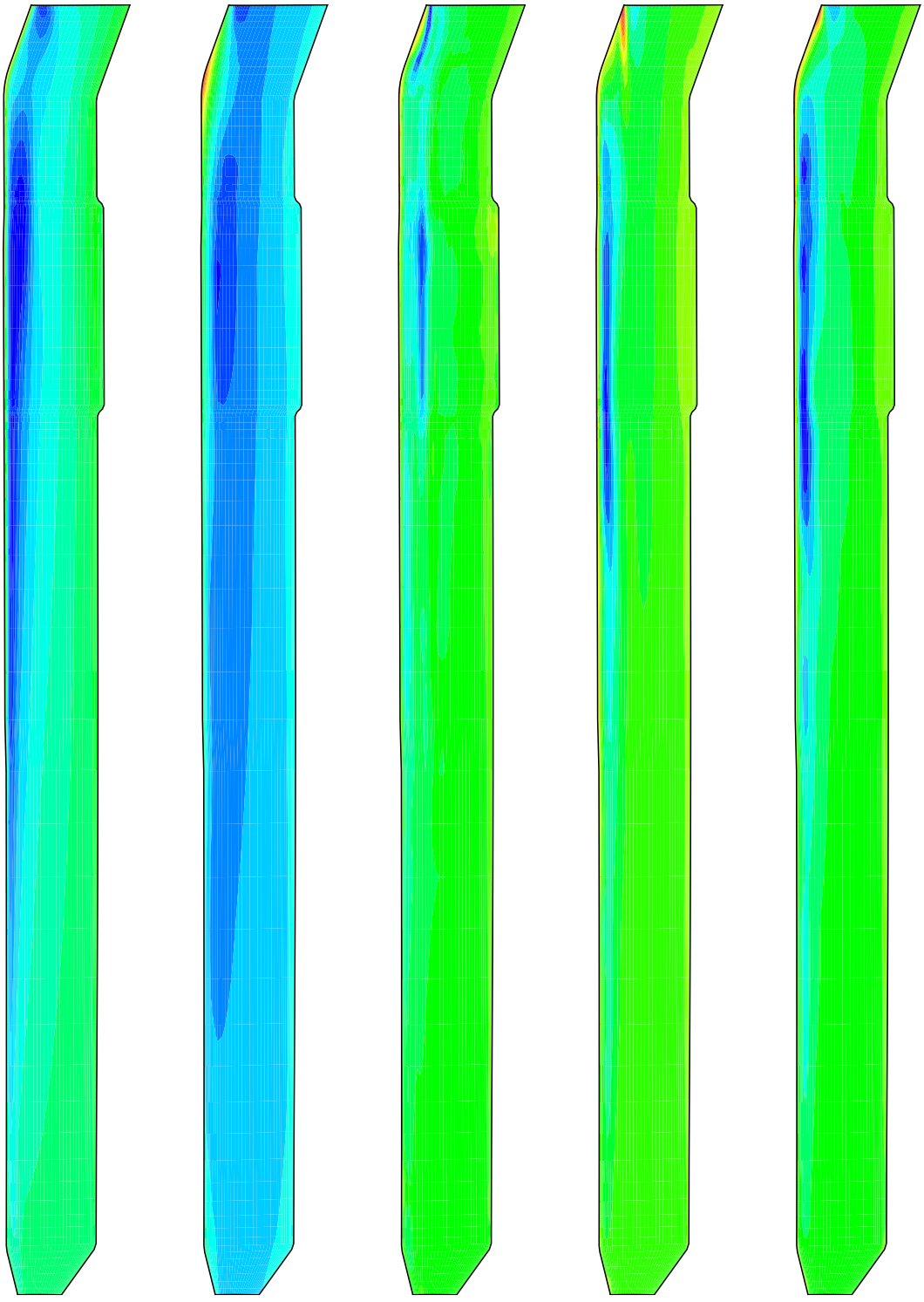
there is noticeable dissipation and the physics of the problem has been altered where the suction peak shrinks in extent and shifts toward the trailing edge. In comparison, the TPS interpolation was able to more accurately preserve the location and extent of the suction peak region on the rotor blade. Near the blade tip where there are relatively high gradients and high grid density, Figure 19 shows a small pressure artifact (red-colored contours) where the interpolated quantity has changed sign over time. However the size of this region is small, and overall after repeated interpolations TPS gives the result closest to the original input solution.

3.4 Investigation of Local Radial Basis Function Properties

In all practical implementations of kernel-based interpolation within computations, only localized interpolations will be performed in the interest of both accuracy and efficiency. When considering overset and/or hybrid simulations, interpolation for potentially millions of locations at each iteration may required, therefore a global interpolation is not feasible due to numerical instability, computation time, and memory requirements.

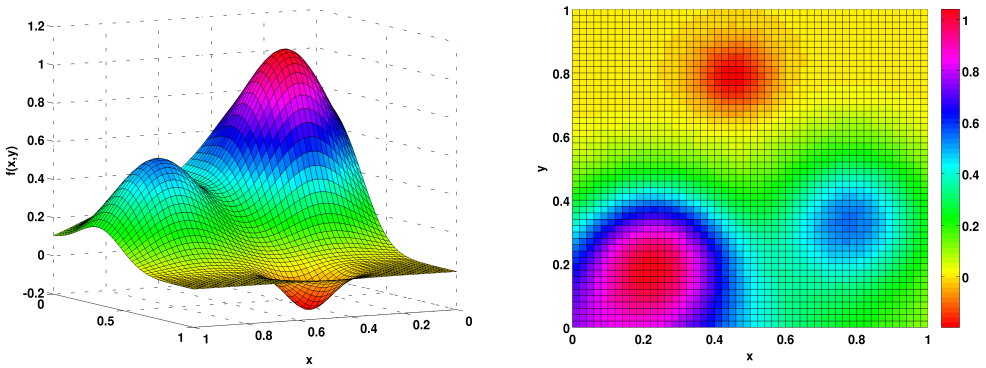
The first test case investigates the behavior of a localized global RBF (e.g., thin-plate splines) compared with actual compactly-supported functions (e.g., the Wendland function). The second case studies the integration of the solution field over individual computational cells. This has major implications for mass conservation, a measure of solution fidelity. If mass or any other conserved quantity changes significantly over time, the physics of the problem are not guaranteed to be correct.

Franke's function (Eqn. 35), a well-known bi-variate function originating from the 1979 work of Richard Franke investigating scattered data approximation [108], has



(a) Input (b) IIM (quad4) (c) IIM (tri6) (d) TPS (e) TPS (local rescaling)

Figure 19: Comparison of the rotor blade solutions after 50 iterations: (a) initial input, (b) inverse isoparametric mapping (IIM) with 4-node quads, (c) IIM with 6-node triangles, (d) TPS with global domain scaled to unity, and (e) TPS with neighborhood scaled to unity.



(a) 3-D View (b) 2-D View

Figure 20: Franke's Function (Eqn. 35) used in the demonstration of compactly-supported functions.

been used as the analytical test function for both cases.

$$f(x,y) = \frac{3}{4}\exp\left(-\frac{(9x-2)^2}{4} - \frac{(9y-2)^2}{4}\right) + \frac{3}{4}\exp\left(-\frac{(9x+1)^2}{49} - \frac{(9y+1)^2}{10}\right) + \frac{1}{2}\exp\left(-\frac{(9x-7)^2}{4} - \frac{(9y-3)^2}{4}\right) - \frac{1}{5}\exp\left[-(9x-4)^2 - (9y-7)^2\right] \tag{35}$$

The function is smooth and qualitatively described by three distinct Gaussian peaks as illustrated in Fig. 20.

When applying a compactly-supported RBF, a support radius was not explicitly specified *a priori* as described in the literature [90, 97]. Instead, coordinates of interpolation source points are translated and scaled such that the points enclosing the target lie within a unit domain centered at zero. Clouds of interpolation points are formed by first identifying the cell enclosing the target point or the source point closest to the target point. Using the cell nodes (or source node) as the starting location, the support region is grown outward by including neighboring nodes and neighbors of neighboring nodes. This approach was chosen for demonstration purposes as it is more efficient than a nearest neighbor search and it minimizes the amount of required code development within an existing data transfer framework. In an actual overset application, a search is necessary since the nearest nodes may not lie on the same

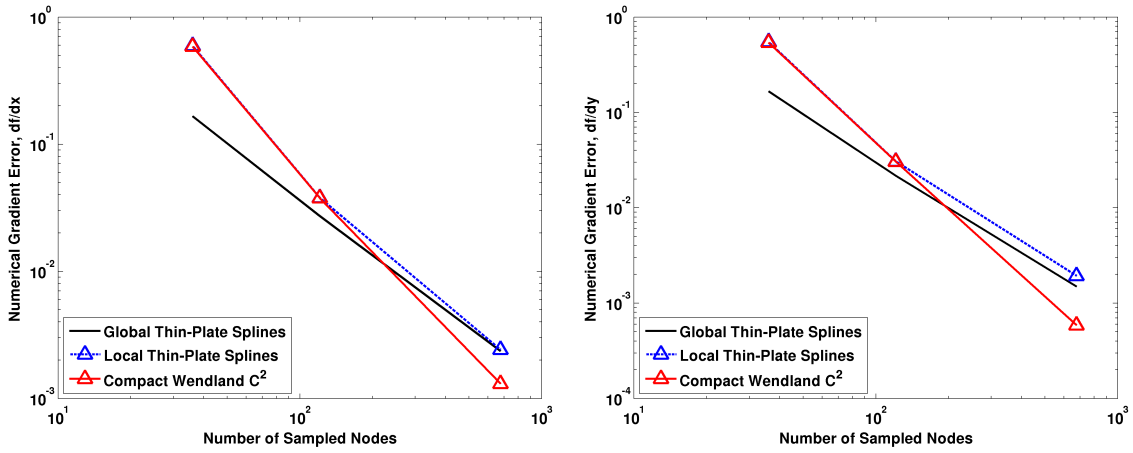
mesh.

3.4.1 Smoothness of Compactly-Supported Functions

This test case is similar to the study of the PDE in Section 3.3.1, in which a source solution is taken from a coarse Cartesian mesh and interpolated onto progressively finer Cartesian meshes. The gradient is then evaluated numerically using second-order central differences. Gradients on the finest mesh (with 2601 points) were calculated for reference. A series of coarse meshes are formed by taking every 2nd, 5th, and 10th point (resulting in sizes of 676, 121, and 36 points respectively) to be used as the input grid. For each point on this mesh, a support region is identified. A point is a support if it lies less than some radial distance from the center. As the domains are unit squares, no rescaling of the input points was performed. The solution is then interpolated from the progressively coarser meshes onto the original fine mesh, with each point on the output mesh using the interpolant centered around the nearest input grid point.

Finite differences of the interpolated solution are then computed and compared to the reference values. Figures 21 and 22 illustrate the resulting gradient errors for support radii of 0.25 and 0.5 based on the demonstration case in Ref. 90. With the smaller region (Fig. 21), both localized RBF approaches produce essentially the same result and are less accurate than the global approach. With increasing number of nodes (i.e., decreasing mesh disparity), the localized TPS results approach that of the global TPS interpolation (supported by all points in the computational domain). However, at the lowest mesh disparity (2 output nodes for 1 input node in each direction) the error associated with the compact interpolation converges steadily and is actually lower than the global interpolant.

For the larger support radius, the local interpolation is essentially equal to the global approach (recalling that the limit of increasing support radius is the entire



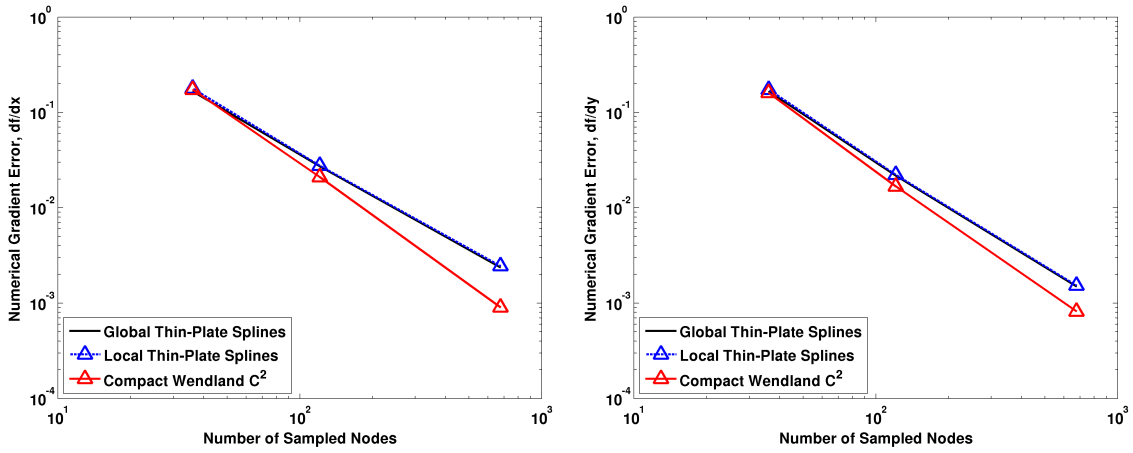
(a) df/dx (b) df/dy

Figure 21: Demonstration of the effectiveness of compactly-supported functions in reproducing gradients of Franke's function (Eqn. 35) using a support radius of $\rho = 0.25$.

input domain, and the local support turns into a global support). The localized and global TPS approaches produce identical results. In comparison the error in the compact Wendland approach starts out equal to the TPS error and then decreases monotonically at a higher rate. These results suggest that in at least some scenarios involving coarse to fine mesh data transfer, compactly-supported functions are able to produce more accurate piece-wise smooth fields.

3.4.2 Conservation

The need to satisfy conservation laws has been discussed in Section 1.1.3, and a number of techniques to mitigate the lack of conservation for overset methodologies have been evaluate in the literature. Zhang et al. [109] demonstrated that with proper selection of interpolation stencils, conservation can be automatically satisfied. Conservative applications of standard interpolation techniques on unstructured meshes have also been demonstrated and recommended by Alauzet et al. [24] based on mesh intersection methods, but extension to three dimensions may be prohibitively expensive [110]. Recently, Florez et al. [111] have demonstrated a mass-conservative RBF



(a) df/dx (b) df/dy

Figure 22: Demonstration of the effectiveness of compactly-supported functions in reproducing gradients of Franke's function (Eqn. 35) using a support radius of $\rho = 0.5$. The global and local TPS solutions are indistinguishable.

interpolation scheme that shows promise for the neighborhood approach. However, Part-Enander & Sjogreen [25] have conversely shown that conservative interpolation can be destabilizing. As an alternative to strict conservation enforcement, the effect of a more accurate interpolation approach on conservation without specific attempts to conserve flow field quantities is investigated.

The RBF interpolant defined in Section 3.1.5 provides a continuous, analytical representation of the original function, and is reproduced here:

$$s_{f,X}(x) = \sum_{j=1}^{N} \alpha_j \Phi(x - x_j) + \sum_{k=1}^{Q} \beta_k p_k(x). \tag{36}$$

This function can then be integrated to provide a high-order analytical expression for the analytical mass

$$M = \int_{\mathcal{H}} u \, \mathrm{d}\Omega, \tag{37}$$

allowing for a more accurate estimate of mass conservation when using higher-order methods:

$$M = \int_{x_1}^{x_2} \int_{y_1(x)}^{y_2(x)} u \, \mathrm{d}y\mathrm{d}x = \int_{\theta_1}^{\theta_2} \int_{r_1(\theta)}^{r_2(\theta)} u \, r \mathrm{d}r\mathrm{d}\theta. \tag{38}$$

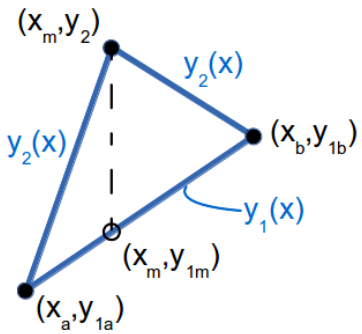


Figure 23: Integration of an arbitrarily oriented triangle in Cartesian space.

When Eqn. 38 is combined with Eqn. 36,

$$M = \sum_i \alpha_i \int_{x_1}^{x_2} \int_{y_1(x)}^{y_2(x)} \phi(r)\,\mathrm{d}y\mathrm{d}x + \int_{x_1}^{x_2} \int_{y_1(x)}^{y_2(x)} (\beta_0 + \beta_1 x + \beta_2 y)\,\mathrm{d}y\mathrm{d}x, \tag{39}$$

where $r \equiv \sqrt{x^2 + y^2}$. While the polar form of Eqn. 38 is a more natural mathematical representation (since the basis function is a function of the radial coordinate), numerical implementation is more straightforward in Cartesian form due to the integration bounds on r. Integration over arbitrarily oriented cells is accomplished by first vertically bisecting the triangle and finding the midpoint of side that will form the new vertical edge (Fig. 23). Evaluation of Eqn. 39 from x_1 to x_2 and $y_1(x)$ to $y_2(x)$ requires selection of the appropriate x coordinate from x_a, x_m, and x_b, and selection of the appropriate $y(x)$ function.

The exact function values at the nodes of a two-dimensional unstructured mesh form the RBF interpolants that were numerically integrated to obtain the mass of each cell. This experiment was performed on three mesh levels of increasing refinement, with 324, 1296, and 5184 cells. These calculations are compared with the linearly integrated mass, which is equivalent to multiplying the cell area by average function value. For the current test function, integration with a higher-order interpolant always results in a more accurate solution (Fig. 24). Since the RBF interpolant is able to recover some of the curvature of the original surface, the volume under this surface

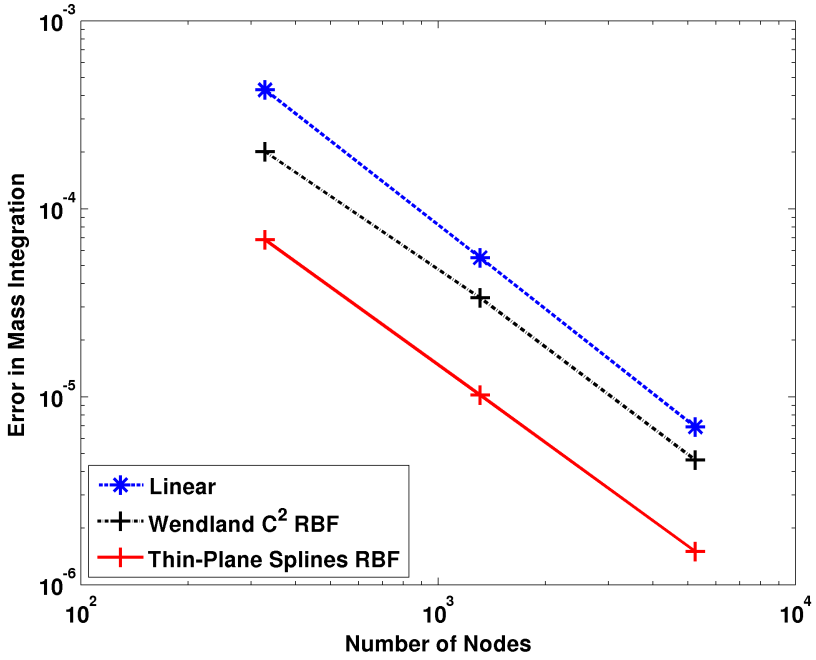


Figure 24: Calculated mass of Franke's function Eqn. 35 by integration of radial basis function interpolants.

is expected to be more accurate. In the case of a localized thin-plate spline RBF, nearly an order of magnitude reduction in error of the mass calculation is achieved in comparison to traditional linear interpolation.

The results depicted in Fig. 24 were obtained by translating each cloud of interpolation points to the origin and then scaling the coordinates so that the difference between maxima and minima in each dimension was one. As the normalization factor increased, the compactly-supported basis becomes increasingly flat due to the normalized radial distances being closer to zero and the function evaluations approaching one for all r. For the nominal scaling (with source points in $[-0.5, 0.5]^2$) both approaches demonstrate consistent improvements over the linear approach.

The TPS approach was relatively insensitive to the normal scaling while the error in the compact W2 approach tended to decrease with larger support regions. On a $[-5, 5]^2$ local domain the errors are comparable to those of TPS (Fig. 25). This has confirmed the work of Song et al. [112] who proved that RBFs of finite smoothness

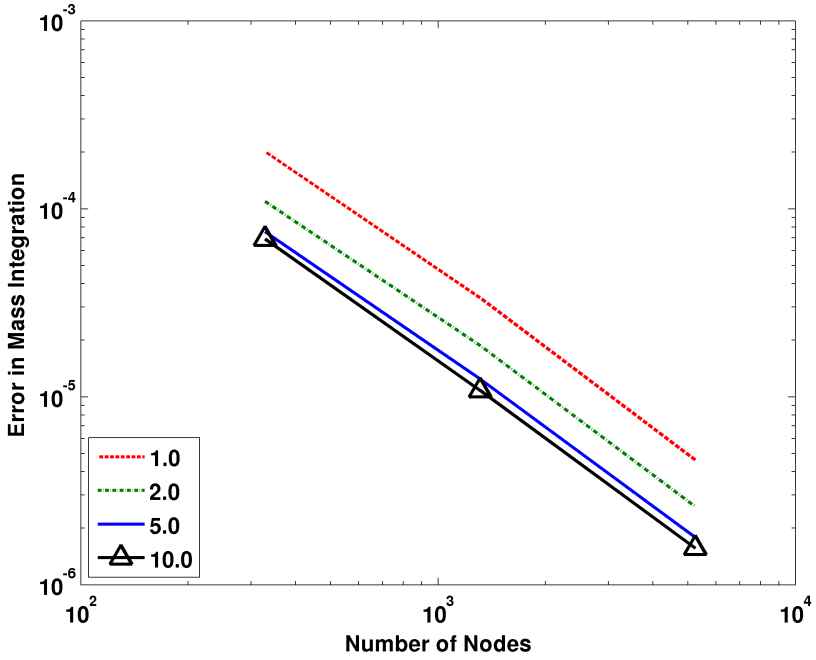


Figure 25: Calculated mass of Franke's function (Eqn. 35) by integration of the Wendland C^2 interpolant for different local re-scalings.

(e.g., the C^2 smooth Wendland function) converge to polyharmonic interpolants (e.g., thin-plate splines) as the function grows increasingly flat.

CHAPTER IV

DATA-ADAPTIVE MAPPING STRATEGIES

The results presented in Chapter 3 have shown that scattered data techniques can be comparable to if not better than higher-order applications of existing techniques (e.g., the inverse isoparametric mapping technique). Moreover, the scattered data techniques afford additional flexibility by removing requirements on sampled data structure, allowing for interpolation and extrapolation of arbitrary grid configurations that may include orphan points. As an alternative to radial basis function interpolation, the IIM approach may also be cast in the form of a scattered data interpolation problem. However, the highest accuracy for the IIM approach is achieved with regularly distributed nodes [71]. The IIM approach is also not suited for extrapolations [77, 73] and the mapping may be non-unique, further limiting applicability. Based on the findings in Section 3.1.3, application of higher-order IIM as a cloud-based approach does not appear feasible due to sensitivities of finite elements to arbitrary point distributions.

Many applications of radial basis function (RBF) techniques to date have the goal of generating complex freeform (i.e., non-symmetric) smooth surfaces. Examples include the design of high-precision optical devices [113] or the generation of topographical mappings from elevation measurements [59]. While much of Chapter 3 has focused on interpolating smooth fields, many aerospace problems especially within the field of fluid dynamics, encounter discontinuous solution fields. In transonic and supersonic flows, the flow field has density, velocity, and pressure discontinuities. Also, when considering the multi-scale, stochastic nature of turbulence, many quantities

may change quickly in both time and space also resulting in discontinuities. Therefore, this chapter investigates the applicability of these techniques to non-smooth solution fields and develops a data-adaptive mapping strategy.

The first section presents a data-adaptive regression technique. This *steering kernel regression* (SKR) method has been adapted from the field of image processing where it has been used successfully in the preservation of edges for image reconstruction and super-resolution problems [87, 114]. Baseline input parameters are established so that improvements over standard linear interpolation may be verified. Afterward, the steering kernel is incorporated into the existing RBF technique.

The second section builds upon the approaches discussed in Chapter 3 and develops a new *anisotropic* kernel interpolation methodology. In this approach, the local solution field dictates the local interpolation support region. Using the steering parameters calculated by the SKR, the local shape, orientation, and extent of the support region are modified based on local solution gradients. The original isotropic kernels (i.e., RBFs) still form the basis for the data transfer but the kernels are now deformed in space to perform a novel *steered radial basis function* (SRBF) interpolation.

The third and fourth sections present results for a number of test functions on two-dimensional structured grids, applying the SRBF technique after establishing best practices. The following chapter presents additional test cases interpolated on unstructured grids, based on similar examples by Alauzet et al. [24] who developed a conservative interpolation method for unstructured meshes. Results have been evaluated based on both interpolation and conservation errors. The final section applies both unsteered and steered scattered data approaches to an actual aerodynamic solution for the turbulent airwake behind a naval ship.

4.1 Steering Kernels

4.1.1 Background

A "steering" scheme has been explored to adaptively optimize any chosen mapping technique. A number of authors have applied the established principal component analysis (PCA) technique for image local orientation estimation [87, 114, 115]. Standard polynomial kernel regression techniques for image restoration and reconstruction rely on local linear combinations of data, with estimates based on sample location and density. Steering techniques modify the local kernel shape and size permitting nonlinear data manipulation and edge preservation [87]. Since this is an approximation technique, exact reproduction of input solution values at input nodes is not guaranteed.

To estimate the local orientation (i.e., the dominant gradients) in a data set, a singular value decomposition (SVD) is performed on a symmetric covariance matrix C_i:

$$C_i = \gamma_i \boldsymbol{U}_{\theta_i} \boldsymbol{\Lambda}_i \boldsymbol{U}_{\theta_i}^T, \tag{40}$$

where the rotation and scaling matrices are defined as

$$\boldsymbol{U}_{\theta_i} = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}, \tag{41}$$

$$\boldsymbol{\Lambda_i} = \begin{bmatrix} \sigma_i & 0 \\ 0 & \sigma_i^{-1} \end{bmatrix}. \tag{42}$$

C_i is estimated from a local gradient matrix G, formed by local subsets of points which can come from overlapping regions. The steering parameters σ_i, θ_i, and γ_i can then be computed for each source point i by singular value decomposition, which determines these parameters in a numerically stable and efficient manner [116]. These parameters correspond to elongation, rotation, and scaling of the kernel, and uniquely specify the transformation [87]. The transformed kernel is represented by a "footprint"

which outlines the region in which the kernel is non-zero (within the support radius). Mathematically, the dominant orientation of the footprint is orthogonal to the gradient vectors on average [116]. This is aligned with the eigenvector corresponding to the smallest eigenvalue of C_i or equivalently the singular vector corresponding to the smallest singular value. The overall approach can be adapted to account for multiple scales and/or noisy data with the inclusion of a multi-scale pyramid decomposition and Kalman filter [116] but this has not been investigated in the present work.

Regression quality depends on the chosen kernel function K with bandwidth h dictating the physical extent of the kernel footprint. In a single dimension

$$K_h(x) = \frac{1}{h} K\left(\frac{x}{h}\right), \tag{43}$$

which may be represented in matrix form as

$$K_H(\boldsymbol{x}) = \frac{1}{|\boldsymbol{H}|} K\left(\boldsymbol{H}^{-1}\boldsymbol{x}\right). \tag{44}$$

for dimensions greater than or equal to one. The steering matrix $\boldsymbol{H}$ as defined in Ref. 87 is:

$$\boldsymbol{H}_i = h\mu_i \boldsymbol{C}_i^{-1/2} \tag{45}$$

where h is a global smoothing parameter, and μ_i is a local density scalar related to the mesh spacing in a neighborhood of the target point. Takeda et al. [87] utilized a Gaussian kernel (a radial basis function):

$$K(\boldsymbol{x}) = \frac{1}{2\pi} \exp\left(-\frac{\boldsymbol{x}^T\boldsymbol{x}}{2}\right) = \frac{1}{2\pi} \exp(-r^2/2). \tag{46}$$

The *steering kernel* is then:

$$\begin{aligned} K_H(\boldsymbol{x}) &= \frac{1}{\det \boldsymbol{H}} \frac{1}{2\pi} \exp\left[-\frac{1}{2}(\boldsymbol{H}^{-1}\boldsymbol{x})^T(\boldsymbol{H}^{-1}\boldsymbol{x})\right] \\ &= \frac{1}{2\pi \det \boldsymbol{H}} \exp\left[-\frac{1}{2}(\boldsymbol{x}^T\boldsymbol{H}^{-2}\boldsymbol{x})\right] \\ &= \frac{\sqrt{\det \boldsymbol{C}}}{2\pi h^2 \mu^2} \exp\left[-\frac{\boldsymbol{x}^T\boldsymbol{C}\boldsymbol{x}}{2h^2\mu^2}\right]. \end{aligned} \tag{47}$$

The definition of the Gaussian kernel in Eqn. 47 is consistent with a non-degenerate multivariate normal distribution [117], i.e.,

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^n |V|}} \exp\left\{ -\frac{(\boldsymbol{x}-\mu)V^{-1}(\boldsymbol{x}-\mu)^T}{2} \right\},$$

in two dimensions ($n = 2$) and centered around zero ($\mu = 0$).

4.1.2 General Solution Procedure

The steering kernel regression procedure described by Takeda et al. [87] has been applied to the present investigations. It is summarized as follows:

1. Estimate the local gradients of the input function to form an estimate of the 2×2 covariance matrix. Gradients are calculated using finite difference algorithms. The compactness of the covariance estimate, i.e., the support radius or the number of neighboring points from which to include gradient information is manually specified. For N source points,

$$C_i \approx G = \begin{bmatrix} \frac{\partial f(\boldsymbol{x}_1)}{\partial x} & \frac{\partial f(\boldsymbol{x}_1)}{\partial x} \\ \frac{\partial f(\boldsymbol{x}_2)}{\partial y} & \frac{\partial f(\boldsymbol{x}_2)}{\partial y} \\ \vdots & \vdots \\ \frac{\partial f(\boldsymbol{x}_N)}{\partial y} & \frac{\partial f(\boldsymbol{x}_N)}{\partial y} \end{bmatrix}. \tag{48}$$

2. Perform an SVD:

$$G = USV^T. \tag{49}$$

   This operation decomposes the local gradient matrix G into three separate matrices. S is a diagonal matrix of singular values, while U and V are the left- and right-hand singular values respectively. V is always a square matrix. These matrices provide the steering parameters that specify the components of the covariance matrix C.

3. Determine the orientation of the local data by calculating the steering parameters defined in Eqn. 40 to 42. The rotation parameter θ is calculated from the right singular vector corresponding to the smallest singular value. For $\boldsymbol{v} = [v_1 \; v_2]^T$:

$$\theta = \tan^{-1}(v_1/v_2). \tag{50}$$

4. Compute the stretching (σ) and scaling (γ) parameters from the singular values (along the diagonal of S):

$$\sigma = s_1/s_2 \approx \frac{s_1 + \lambda'}{s_2 + \lambda'}, \tag{51}$$

$$\gamma = \sqrt{s_1 s_2} \approx \sqrt{\frac{s_1 s_2 + \lambda''}{N}}. \tag{52}$$

   N in this case is defined as the number of source points applied to the steering calculation. Very small values of s_2 can give $\sigma \to \infty$ (indicative of poor conditioning in the local gradient matrix due to very large gradients) and $\gamma \to 0$. To mitigate these issues, regularization parameters ($\lambda' = 1.0$ and $\lambda'' = 0.01$) have been specified [87].

   For numerical stability, the γ steering parameter has been simplified to $\gamma_i = \sqrt{\frac{\lambda''}{N}}$. This simplification is equivalent to assuming that the singular values from the SVD are both equal and approximately zero, corresponding to the degenerate case of an SVD performed on a local gradient matrix of all zeroes (describing a constant solution field).

5. The approximate covariance matrix is calculated from Eqn. 40, which modifies the input displacements to the kernel function (per Eqn. 47). This changes the weighting at each source location.

6. The steering kernel-assisted regression is performed by solving the weighted optimization problem,

$$\hat{\boldsymbol{b}} = \arg\min_b ||\boldsymbol{y} - X_x \boldsymbol{b}||^2_{W_x}, \tag{53}$$

for linear regression coefficients b, with the linear algebra solution being:

$$\begin{aligned} X_x\hat{\boldsymbol{b}} &= \boldsymbol{y} \\ W_x X_x\hat{\boldsymbol{b}} &= W_x\boldsymbol{y} \\ X_x^T W_x X_x\hat{\boldsymbol{b}} &= X_x^T W_x\boldsymbol{y} \\ \hat{\boldsymbol{b}} &= (X_x^T W_x X_x)^{-1} X_x^T W_x\boldsymbol{y}. \end{aligned}$$

The input matrices and vectors are:

$$X_x = \begin{bmatrix} 1 & (\Delta r_1)^T & \cdots \\ 1 & (\Delta r_2)^T & \cdots \\ \vdots & \vdots & \\ 1 & (\Delta r_N)^T & \cdots \end{bmatrix} \tag{54}$$

$$\Delta r_i = \begin{bmatrix} (x_i - x_o) & (y_i - y_o) \end{bmatrix}^T \tag{55}$$

$$W_x = \text{diag}\begin{bmatrix} K_H(\Delta\boldsymbol{r}_1) & K_H(\Delta\boldsymbol{r}_2) & \cdots & K_H(\Delta\boldsymbol{r}_N) \end{bmatrix} \tag{56}$$

$$\boldsymbol{y} = \begin{bmatrix} f(\boldsymbol{x}_1) & f(\boldsymbol{x}_2) & \cdots & f(\boldsymbol{x}_N) \end{bmatrix}^T \tag{57}$$

$$\hat{\boldsymbol{b}} = \begin{bmatrix} \beta_0 & \beta_1 & \beta_2 \end{bmatrix}^T \tag{58}$$

$$\boldsymbol{e_1} = \begin{bmatrix} 1 & 0 \end{bmatrix}^T \tag{59}$$

with (x_o, y_o) indicating the target location. The elements included in the matrix X_x (Eqn. 54) dictate the order of the regression. With only ones, Δx, and Δy included, the estimator is linear [87].

Since the estimate of $f(\boldsymbol{x})$ is centered at the output node, the first order (and higher) $\hat{\boldsymbol{\beta}}$ terms will always evaluate to zero at the location of interest. The approximate solution is:

$$f(\boldsymbol{x}) \approx \boldsymbol{e}_1^T (X_x^T W_x X_x)^{-1} X_x^T W_x \boldsymbol{y}. \tag{60}$$

4.1.3 Steering Options

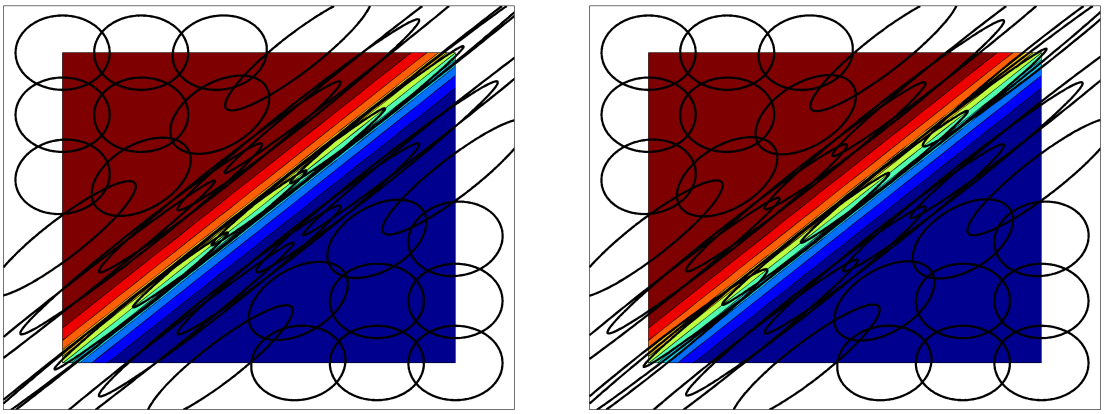
Some of the inputs and sensitivities in the steering kernel approach are discussed in the following subsections. These include a discussion of gradient estimate requirements as well as user-specified parameters for the regression. Evaluations were performed for a simple two-dimensional test case modeling a continuous, differentiable step change along $y = x$ on the unit square domain, defined mathematically as

$$f(x, y) = -0.5 \tanh\left(\frac{x - y}{s}\right) + 0.5. \tag{61}$$

Gradient Estimation

For structured grid topologies, second-order central differences have been applied on the interior of structured domains, and second-order one-sided differences were applied at the boundaries. For unstructured grid topologies (e.g., triangular faces in two dimensions), gradient estimation is not necessarily straightforward. For example, even the simplest higher-order techniques will require the projection of values from neighboring triangles, and the standard approach is to use Green-Gauss integration [118]. For demonstration purposes, a first-order approximation to the gradients is obtained assuming that the function on each mesh face is piecewise planar. The partial derivative of the equation representing this plane in three-dimensional space estimates the function gradients at cell centers. To recover gradients at nodal locations, the gradients of all cells surrounding a given node are area-weighted and summed together. Higher-order differences produce more accurate gradient estimates but did not have an appreciable effect on the final interpolated solution. The reason for this is that the SVD determines the aggregate effect of a number of local source points ($N \sim \mathcal{O}(10)$). Therefore, high precision is not absolutely necessary if the gradient estimates can capture the correct local trends. Moreover, when implemented within a fluid dynamics methodology the solver itself would provide the most accurate gradient estimate in practice.

To evaluate the sensitivity of the steering kernel scaling to numerical errors in the gradient approximation, the analytical gradient of an input function approximating a discontinuous field (Eqn. 61) has been evaluated for input into the steering procedure. This approach results in the behavior illustrated in Fig. 26a, showing proper elongation of the kernel footprints and alignment with the discontinuity near to the high-gradient region. The same steering parameters were calculated based on second-order finite-difference approximations to the input function (Fig. 26b) which demonstrated minimal sensitivity to the gradient estimation technique. The only noticeable differences are in the footprints near the discontinuity, which have a kernel elongation error of less than 2% for the finite-difference approximation. Since the variations are minimal even for the present case with relatively large gradients, higher-order gradient estimation algorithms have not been considered.



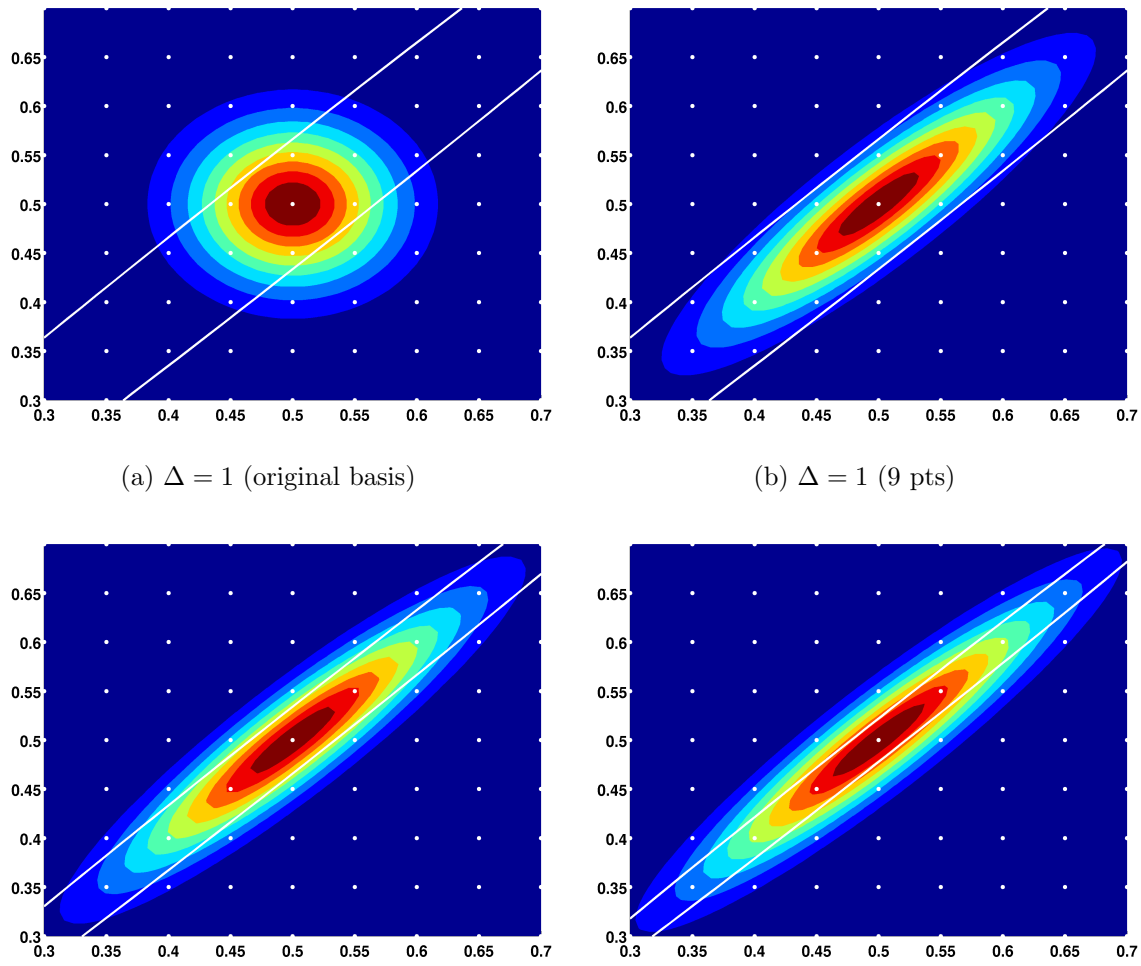
(a) Analytical gradients

(b) Second-order finite differences

Figure 26: Steering kernel footprints computed with exact and approximate gradients of the input function for a smooth step change (Eqn. 64).

Steering Region Size

For the current evaluation case on a structured grid, the effect on the steering transformation of increasing the interpolation neighborhood size is illustrated in Fig. 27.



(a) $\Delta = 1$ (original basis)

(b) $\Delta = 1$ (9 pts)

(c) $\Delta = 2$ (25 pts)

(d) $\Delta = 3$ (49 pts)

Figure 27: Calculated kernel footprint for various steering region sizes, overlaid on contours of the transformed Wendland C^2 radial basis.

On this structured grid, an interpolation cloud can be described by a Δ value indicating the number of neighboring nodes to include in each direction. This configuration resulted in $(2\Delta+1)^2$ total source points in two dimensions. As Δ is increased, the kernel footprint becomes more compact (Fig. 27b–d) which has two consequences. First, the overall size of the footprint is reduced meaning that fewer points (if any) across a discontinuity or edge will be weighted in the interpolation, improving accuracy. Second, there will also be fewer points included in the direction along a discontinuity contributing to the interpolation, which may adversely affect accuracy. In practice, the ideal window size may be problem and grid-dependent.

An additional consideration is the complexity of the flow field. In the current case, a single discontinuity exists in the solution; for a realistic application with turbulent three-dimensional flow, more varied kernel elongation and orientation are expected from many flow features. To avoid agglomerating the effect of multiple flow features, smaller window sizes should be used. However, for $\Delta = 1$, the kernel footprint is not sufficiently stretched to properly exclude the diagonal of points above and below the $y = x$ (Fig. 27). Therefore, a nominal window size of $\Delta = 2$ (with 25 points) was specified for this structured grid case.

For unstructured grids, it is more natural to expand the interpolation cloud outward starting from a single cell rather than a single point. Adding points to a given steering support entails including levels of neighboring nodes connected by edges to the cell under investigation. In the initial effort, only a single expansion was performed forming clouds of 11-12 points surrounding each cell on average.

Interpolation Neighborhood Size

The size of the interpolation neighborhood is independent from the steering window size. In general, the more information included in the mapping, the greater the potential for accuracy but at greater cost. Even though an interpolation is always

numerically stable when two or more distinct points are used [119], an interpolation based on a single cell generally affords no accuracy improvements over bilinear interpolation as demonstrated in Section 3.3.1. To standardize the approach for structured and unstructured grids (since specifying a support radius or a Δ value does not consistently provide the same number or location of points), the same points selected for the steering parameter calculation were also utilized as source points for the data transfer.

Global Smoothing Parameter, h

This parameter described in Ref. 87 controls the shape of the Gaussian kernel and is related to the variance of the distribution. In RBF terminology this is analogous to the shape parameter, some examples of which are listed in Table 1 of Section 3.1.5. From an image processing perspective, this quantity is specified with the assumption that the sampled data is relatively free from noise which is not true for turbulent quantities. To limit the smoothing of the sampled field (i.e., dissipation or smearing in the resolved field), this value has been set to a minimal value. However, excessively small values caused numerical instability, and optimal h was found to be grid dependent. Based on these restrictions, the global smoothing parameter has been set to the mesh spacing as this is the smallest distance between locations at which solutions are known.

Local Density Parameter, μ_i

An iterative approach based on typical statistics and data analysis techniques is employed by Takeda et al. [87] to estimate this value. Density estimation remains an active area of research with a number of algorithms available [120], but these were not applied in the current work to avoid additional solution uncertainties. By analogy to the global smoothing parameter (Eqn. 47), larger values of this parameter have a smoothing effect. Since smoothing is not desirable for fluid dynamics applications,

the local density was iteratively minimized. Starting with a nominal value of one, the parameter was lowered until observed regression error was a minimum which occurred for $\mu_i = \mu \sim h$. For simplicity, μ is also set to the mesh spacing.

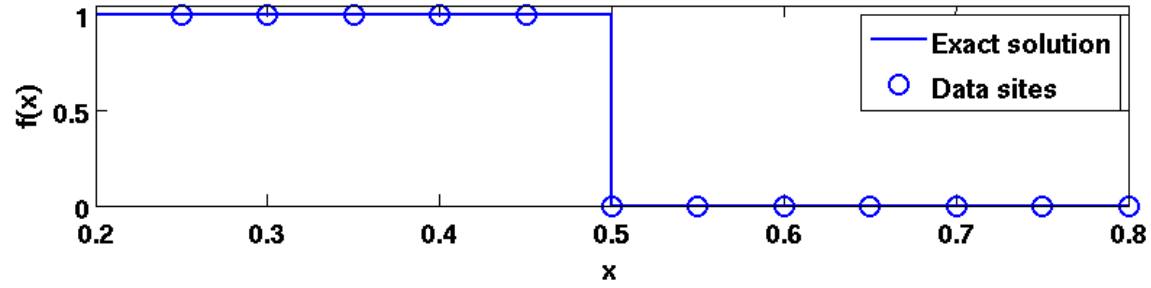
4.2 Steered Kernel Interpolation

The radial basis (RBF) approach explored in Chapter 3 and the steering kernel (SKR) approach introduced in Section 4.1 provide the framework for a robust data-adaptive solution transfer technique. Instead of applying the steering kernel for weighting purposes, the elongation and rotational transformation components from Ref. 87 have been applied to form an anisotropic kernel function from a radial basis. In effect, this transforms a radial function into an elliptical function. The described steering approach is equivalent to remapping or deforming the region within the support radius. As a result, the steered radial basis function (SRBF) evaluates a new set of points, effectively transforming the original function.

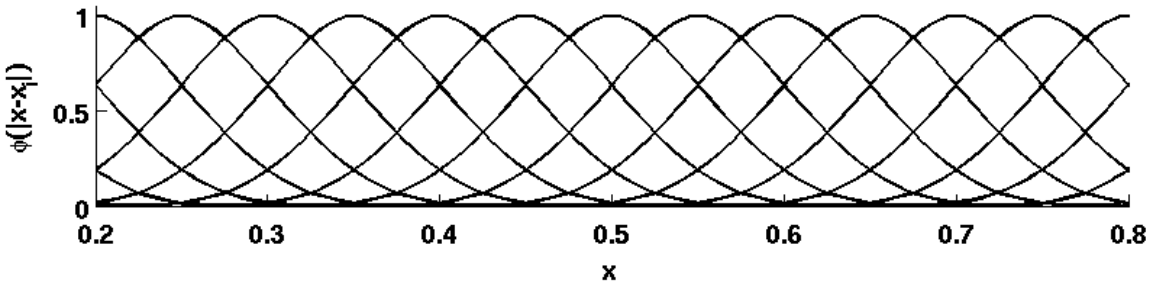
4.2.1 Procedure

This section presents a novel approach to systematically adapt radial basis function interpolation based on the steering kernel regression technique presented in Section 4.1. As an introductory example, Figure 28a illustrates a one-dimensional step function along with the discrete solution that provides source points for the interpolation. Since the RBF centers are chosen to coincide with the locations at which the solution is known (discussed in Section 3.1.5) the isotropic kernel functions are plotted at the same locations in Fig. 28b.

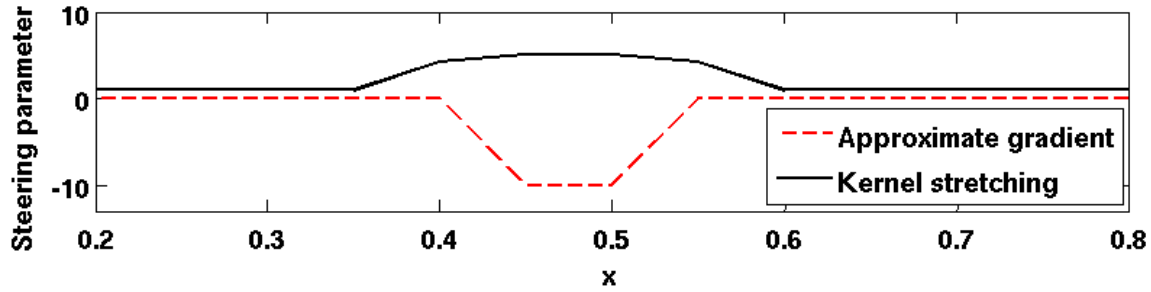
Calculated steering parameters encapsulate information regarding local solution behavior. The kernel stretching parameter is estimated by the singular values of the covariance matrix (Eqn. 51), which is populated by gradient estimates determined by finite differences (as outlined in Section 4.1.2). The stretching parameter operates in the direction normal to rather than in the direction of the dominant gradient.



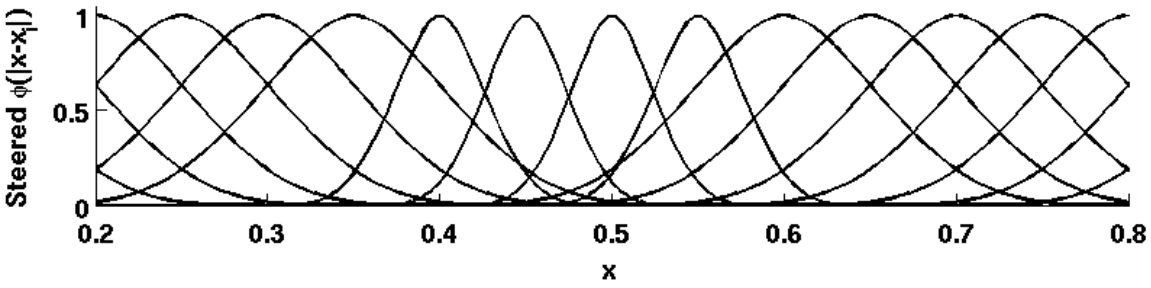
(a) Step function



(b) Wendland C^2 kernel



(c) Steering parameter



(d) Steered Wendland C^2 kernel

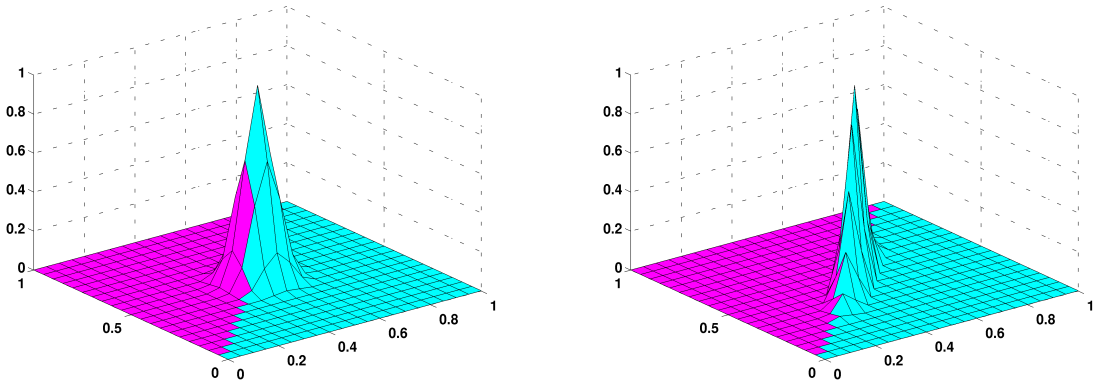
Figure 28: Comparison of steered and unsteered kernel functions applied to RBF interpolation of a one-dimensional step function.

This is more easily visualized in two dimensions (Fig. 26). Since the gradient of the exact step function is discontinuous, the discrete approximation is distributed over two grid points (Fig. 28c). Since the steering parameters are also determined from a neighborhood of gradient estimates (Eqn. 48), the effects of the discontinuity on kernel stretching are further distributed. The basis function is then adapted by transforming the independent variable with steering parameters. From the Gaussian steering kernel defined by Takeda et al. [87] (Eqn. 47), the radial coordinate is related to the covariance matrix as $r^2 = \boldsymbol{x}^T C \boldsymbol{x}$, where $\boldsymbol{x}$ is a position vector. An equivalent radial vector can then be calculated:

$$\begin{aligned} r^2 &= \boldsymbol{r}^T \boldsymbol{r} \\ &= \boldsymbol{x}^T \boldsymbol{C} \boldsymbol{x} \\ &= \gamma \boldsymbol{x}^T \boldsymbol{U}_\theta \boldsymbol{\Lambda} \boldsymbol{U}_\theta^T \boldsymbol{x} \\ &= \gamma (\sqrt{\boldsymbol{\Lambda}} \boldsymbol{U}_\theta^T \boldsymbol{x})^T (\sqrt{\boldsymbol{\Lambda}} \boldsymbol{U}_\theta^T \boldsymbol{x}) \\ \boldsymbol{r} &= \sqrt{\gamma \boldsymbol{\Lambda}} \boldsymbol{U}_\theta^T \boldsymbol{x} \end{aligned} \tag{62}$$

In one dimension, the stretching parameter on the diagonal of Λ (Eqn. 42) scales the radial coordinate, which has the same effect as scaling the support radius (the value at which the basis function becomes identically zero). The reduced support radii near the discontinuity are illustrated in Fig. 28d.

In two dimensions, the support radius loses meaning as the stretching parameter (σ) operates differently on the coordinates in each dimension. Therefore, the resulting kernel function is no longer radial, as depicted in Fig. 29. In addition, non-zero gradients in the second dimension introduce the possibility of kernel rotation. The extent of the kernel footprint (where the basis function is non-zero) is described by an ellipse, with semimajor and semiminor axes scaled by σ and $1/\sigma$, respectively. While the physical extent of the kernel footprint remains constant (the area of the elliptical support is $\pi(\sigma R)(R/\sigma) = \pi R^2$), the number of source points within this region is



(a) Wendland C^2 radial basis

(b) Steered Wendland C^2 elliptical basis

Figure 29: Comparison of unsteered and steered kernel functions at a discontinuity for a step change across $y = x$.

liable to change especially with non-uniform grids.

The transformations move the source points that should not contribute to the interpolation (e.g., points across a shock front) *towards* the shock front but to a greater distance *away* from the RBF center (located at coordinates 0.5,0.5 in this case) as illustrated by the arrows in Fig. 30. At the same time, points that are along the shock front are moved closer to the RBF center, the net effect being that the radial distances for points along the shock are reduced and thus more strongly weighted in the interpolation problem. Where the local solution field is relatively smooth, the kernel is isotropic and the transformation reduces to multiplication by the identity matrix.

Since undesirable points should have a lower weight or RBF value, the compactly supported Wendland C^2 RBF has been chosen instead of the more common thin-plate splines for the steered interpolation approach because of its monotonic behavior on the unit domain (Fig. 31). By definition, a compactly-supported function vanishes outside of a selected support radius. The thin-plate spline has a value of zero at $r = 1$ but is non-zero for $r > 1$. For comparison, the Gaussian RBF (with a standard deviation of 0.5) has also been plotted. While the behavior is similar to the Wendland

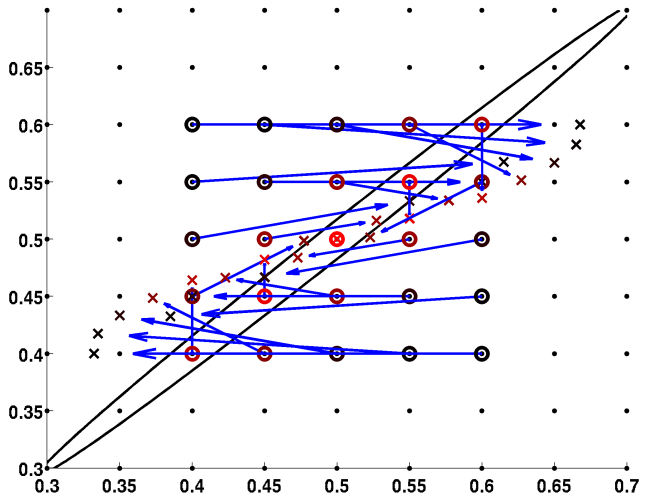


Figure 30: Effect of steering transformation on interpolation points near an edge along $y = x$, with o's indicating the original location of sampled points and x's indicating the transformed coordinates.

C^2 function, it only approaches zero and is non-zero for all $r > 0$. The monotonicity condition ensures that the steering behavior has a consistent effect, since the steering transformation will always increase the radial distance of data points that should be less heavily weighted (e.g., for points that span large gradients). Therefore, the chosen function should always evaluate to lower values at larger radial distances.

The resulting steered basis functions more heavily weight data along rather than across high-gradient regions. This forms a powerful data-adaptive interpolation technique that can more accurately reproduce discontinuities in a solution with minimal overshoot or Gibbs-type behavior. A natural aerodynamics example is the solution of a shock wave. The interpolant should depend on source points that vary in number and location, based on the local solution field. In the vicinity of the shock, the interpolation method should ideally only interpolate from one side of the shock so that the interpolation input function specifies a more favorably smooth function. This idea is analogous to well-established upwind numerical schemes which are designed

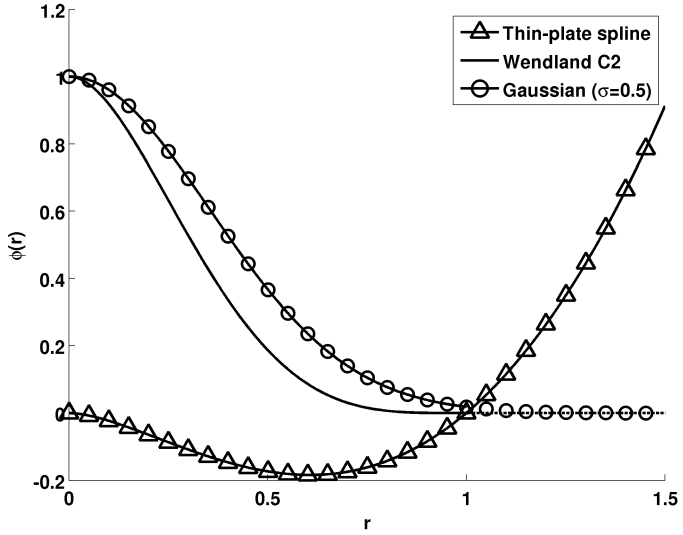


Figure 31: A comparison of selected radial basis functions, normalized by support radius. For $r \geq 1$, the compactly-supported Wendland family of functions is identically zero by definition.

to avoid the same situation [62]. Therefore, the steering algorithm can also be considered a scattered-data limiter which enable variable levels of resolution with the objective of increasing solution fidelity near solution discontinuities. Alternatively, to accomplish a degree of data-adaptivity, Mak and Kung [121] have successfully used covariance data to augment RBF network methods for the purposes of categorizing remote sensing images. Luo et al. [88] have applied an elliptical basis approach with an expectation maximization algorithm to estimate additional elliptical parameters. However, the steering kernel (Eqn. 47) inspires a simpler technique that can achieve the same anisotropic properties without estimation of many parameters thus reducing uncertainty.

4.2.2 Filtering of Source Points

To further increase accuracy and efficiency, a high-pass filter is imposed on the interpolation weights, making it possible to eliminate extraneous source points from the interpolation. When used in combination with the SRBF technique, the following improvements are possible:

- Numerical accuracy is increased because fewer interpolation points implies a smaller, better-conditioned interpolation matrix.

- A smaller interpolation matrix is faster to invert, increasing computational efficiency.

Table 4: Condition numbers for the interpolation of the two-dimensional step function, with and without a steering strategy.

	Average	Condition Number		
	Source Pts.	avg	min	max
Unsteered, no filtering	25	378	376	390
Steered, no filtering	21.5	202	12.4	288
Steered, filtered ($\phi < 0.1$)	16.3	193	12.4	266
Steered, filtered ($\phi < 0.2$)	9.68	187	12.4	260

The filter requires a single user-specified cut-off level. For the two-dimensional step function test case, the original unfiltered neighborhood included 25 source points. Application of the steering strategy described in the previous section had the most pronounced effect on the condition of the interpolation matrix (Eqn. 28). On average the condition number was reduced in half and the minimum condition number reduced by an order of magnitude (Table 4). The steered case without filtering included less than 25 points because the steering transformation can translate some points outside of the support region. Filtering the source points did not have an observable impact on performance for this problem, but did further reduce the average and maximum condition number by 4 and 8%, respectively. Increasing the cut-off value further reduced the average and maximum condition number while the minimum value remained constant. For the remaining investigations in this chapter, the nominal value of 0.1 has been selected. This eliminated the source points at a radial distance greater than 0.58 when applying the W2 function (Fig. 31). For problems studied on a Cartesian grid, this eliminated over 30% of source points on average; on

unstructured grids, over 70% of source points on average were removed. The reason that more points are removed in the unstructured case is because the support regions formed from triangles tended to be more irregular in shape, resulting in more outlying points. For degenerate cases where the filtered support region had very few points, for example less than the number of points defining a single cell that would have been utilized in a linear approach, the original set of source points has been applied.

4.2.3 Algorithm

In summary, the steered kernel regression procedure is as follows:

1. Estimate derivatives at all source points within a specified neighborhood of the target point to form a local gradient matrix.

2. Perform an SVD on the matrix to obtain elongation and rotation parameters (Eqns. 51 and 50).

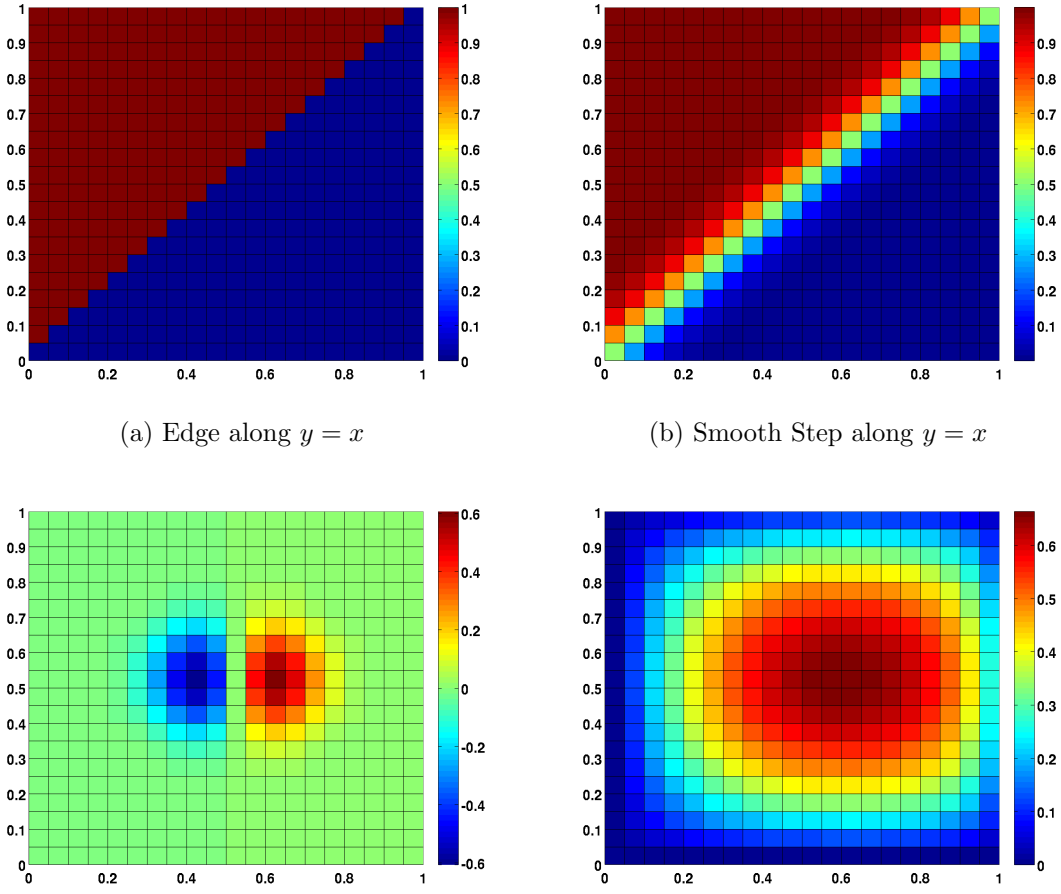
3. Transform the spatial coordinates of all source points.

4. Evaluate the kernel function at the transformed source points and remove points that have close to zero weighting.

5. Apply remaining transformed points to the interpolation problem defined by Eqn. 28.

4.3 Interpolation Tests

The steering kernel approach was evaluated with four closed-form solutions on a 21×21 unit square domain with identical and uniform spacings in all directions (Fig. 32). Functions were transferred onto a coarser mesh, and the input/output mesh disparities are shown in Fig. 33. These cases represent simple examples of different types of gradients that may be encountered in flows of interest.



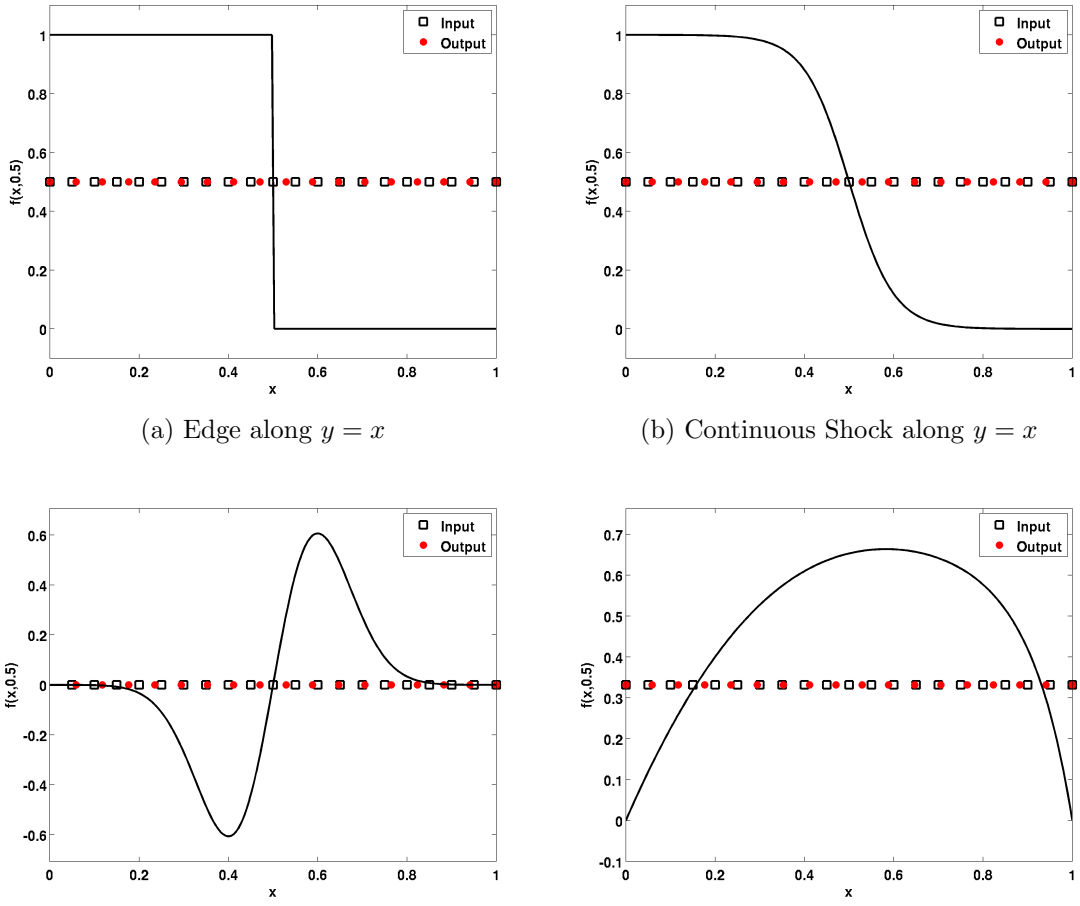
(a) Edge along $y = x$

(b) Smooth Step along $y = x$

(c) Inviscid Vortex

(d) Elliptic PDE

Figure 32: Steering kernel test input domains.



(a) Edge along $y = x$

(b) Continuous Shock along $y = x$

(c) Inviscid Vortex

(d) Elliptic PDE

Figure 33: Steering kernel test functions and mesh spacings on fine and coarse domains (21×21 and 18×18 respectively), sampled through the center of computational domain at $y = 0.5$.

Two high-gradient solution fields were studied. First, a discontinuous step change representative of an exact shock feature in fluid dynamics, was modeled as:

$$f(x,y) = \begin{cases} 1, & y > x \\ 0, & y \leq x \end{cases} \tag{63}$$

Second, a smooth step change was investigated, approximating a shock solution from a slope-limited numerical solution. This continuous solution is modeled as

$$f(x,y) = -0.5 \tanh\left(\frac{x-y}{s}\right) + 0.5. \tag{64}$$

The solution field approximates a discontinuous step change across $y = x$, with the maximum gradient determined by the value of s. This has been set to a nominal value of 0.1 corresponding to a maximum gradient of $\dfrac{\partial f}{\partial x} = 10$. Next, two relatively smooth, low-gradient solutions were investigated. An inviscid vortex was studied as detailed in Section 5.2. Finally, a solution for a PDE subject to Dirichlet boundary conditions was considered as in Section 3.3.1. Bilinear and steering kernel regression (SKR) approaches are compared with thin-plate spline (TPS), Wendland C^2 (W2), and steered Wendland C^2 (W2) RBF approaches.

4.3.1 Results

Table 5 shows the norm of absolute error when mapping from a fine 21×21 domain onto a coarse 18×18 domain. The steering kernel regression approach shows significant improvement over linear interpolation for cases with large gradients in the solution, reducing error by over 40% for the smooth step and by nearly an order of magnitude for the edge case. In the case of the edge, the SKR had the lowest error out of all methods and outperformed the next-best steered RBF technique by nearly 70%. In the absence of strong gradients however, the SKR reduced to linear accuracy.

In all cases, the RBF approaches outperform the linear interpolation, by over 50% in the edge test case, and by approximately an order of magnitude in all other

Table 5: Error norms for select test functions and mapping techniques

Test Function	Linear	SKR	TPS	W2	SW2
Edge	0.3667	0.0545	0.2589	0.2054	0.1660
Smooth Step	0.0490	0.0275	0.0043	0.0034	0.0027
Isentropic Vortex	0.1285	0.1243	0.0098	0.0101	0.0102
Elliptic PDE	0.0387	0.0473	0.0067	0.0076	0.0099

cases. The TPS basis achieved the lowest error in the low-gradient cases (vortex and elliptic PDE solutions), with up to 12% reduction in error over the standard W2 implementation. Applying the steering technique to the W2 function reduced errors by 20%, outperforming the TPS approach. The steering approach was not effective in the isentropic vortex case (with nearly identical error compared to the standard W2 implementation), and was worse than W2 for the elliptic PDE. However, in this case the regression technique produced worse results than the linear approach, which was also reflected in the steered RBF approach.

Table 6: Interpolation quality metrics for an interpolated step change between 0 and 1; the finite slope at the discontinuity is -20.

Interpolation Method	Min	Max	Max Over/ Undershoot [%]	Slope	Slope Error [%]
Bilinear	0	1	0	-6.9135	65.4
Steering kernel regression	0.0000	1.0000	0	-17.005	15.0
Thin-plate spline	-0.0645	1.0804	8.04	-10.838	45.8
Wendland C^2	-0.1068	1.1225	12.3	-12.102	39.5
Steered Wendland C^2	0	1.0124	1.01	-18.687	6.57

Figure 34 offers insight into the effect of applying a steering strategy. While a typical bilinear interpolation will evenly smear the gradient in two directions, the error in the steered approach is directed toward one side. The RBF approaches minimize error by preserving the gradient at the discontinuity at the expense of Gibbs-like over- and under-shoot, for example 8% with the TPS basis and 12% with

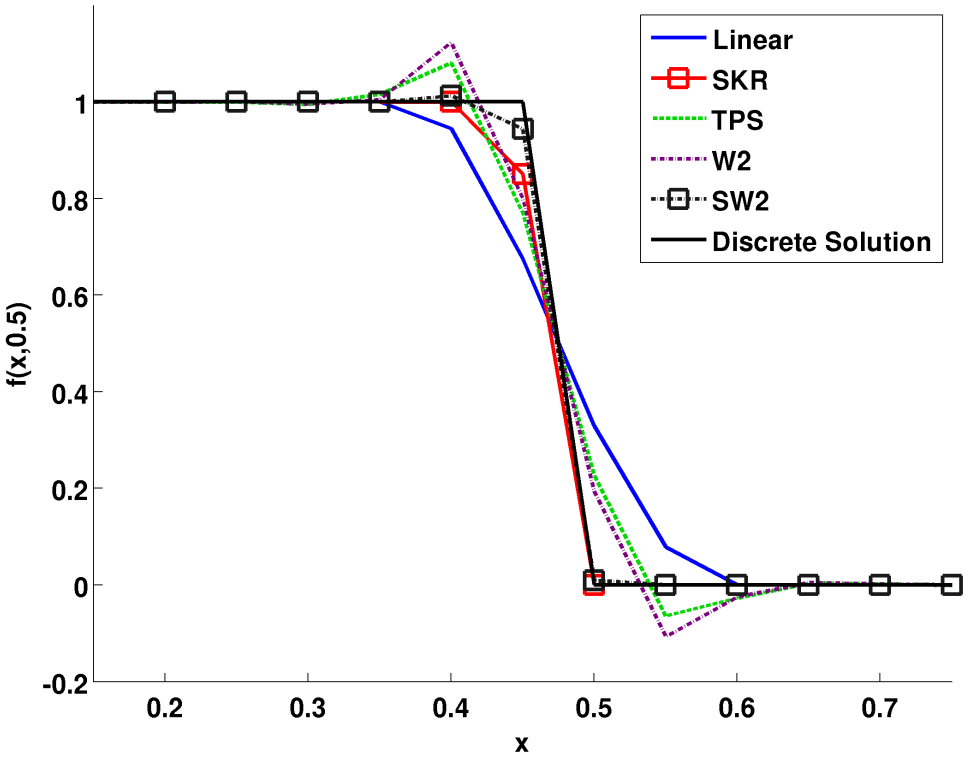


Figure 34: Solution for a two-dimensional step function after a single interpolation from a coarse grid to a fine grid for all mapping approaches. The discrete solution described by Eqn. 63 has been plotted for reference.

the W2 basis (Table 6). While the steered solution is not completely monotone (from the perspective of a total variation diminishing limiter), the error is successfully mitigated with over- and under-shoot reduced to only 1%. In addition, the steered W2 interpolation reduced the slope error at the discontinuity by 33% over the unsteered W2 approach, with only 7% error. In comparison, the SKR approach has no over- and under-shoot but underpredicts the slope by 15%.

4.3.2 Convergence

While Section 3.3.1 quantified the rate of convergence for a smooth solution field, interpolation errors evolve differently within discontinuous fields. The two-dimensional step change function studied in this section (Eqn. 63) was considered with increasingly refined source grids, containing 21×21 to 321×321 points. Resulting errors are plotted in Fig. 35. The steering kernel regression approach applies the Gaussian kernel which depends on a number of tunable input parameters; for this convergence study, the parameters were kept constant based on the development in Section 4.1.3 and caused the left-hand-side matrices in the regression solution to become poorly conditioned. As a result, the SKR solution diverged, as indicated by the upward trend in the curve for $\Delta s < 0.0125$. The calculated steering parameters do not depend on the input parameters and therefore did not affect the accuracy of the steered RBF approach.

Both the steered approaches (steering kernel regression and steered Wendland C^2 interpolation) initially converged at an approximately second-order rate (Table 7). The linear and unsteered RBF methods converged at a much slower rate, with a slope of approximately 0.6. Moreover, the errors were nearly constant for $\Delta s < 0.25$. In comparison, the steered Wendland C^2 solution had over three orders of magnitude less error than the linearly interpolated solution at the finest grid level. This approach was also able to maintain convergent behavior comparable to the initial convergence

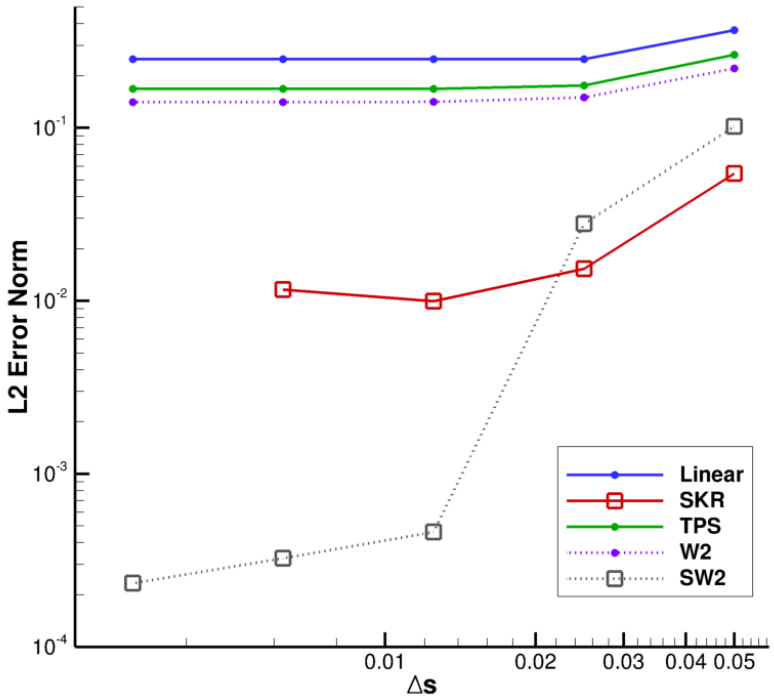


Figure 35: L2-norm of the interpolation error for the two-dimensional step function described by Eqn. 63.

rate of the other RBF approaches.

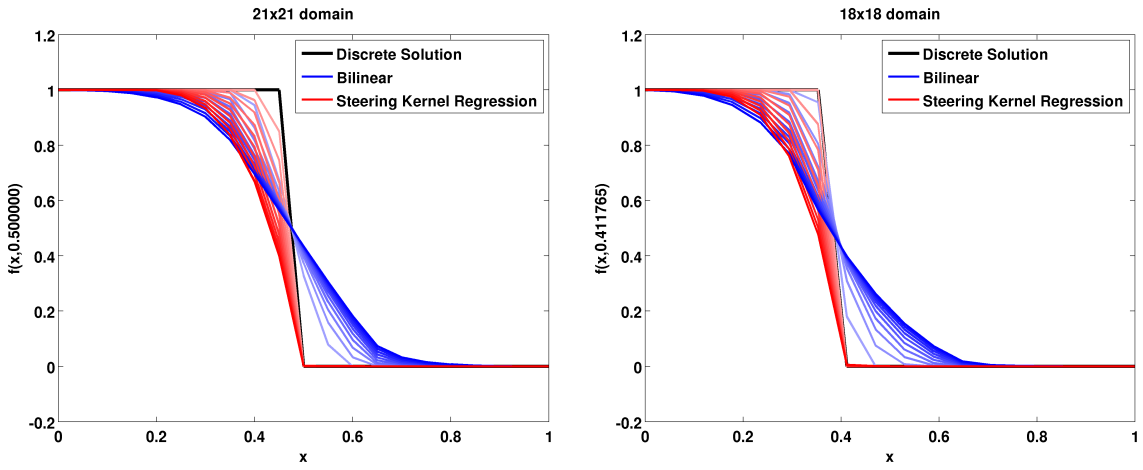
4.4 Repeated Interpolation Test

Results for an interpolated step function described by Eqn. 63 are illustrated in Figs. 36 and 37 for linear and RBF-based approaches. As with the rotor blade case in Section 3.3.2, the solution has been repeatedly transferred starting from a fine grid (21×21) onto as coarse grid (18×18), and then back onto the original fine grid (constituting a single iteration). This test emulates the effects of repeated overset information transfer in a time-accurate overset solution and demonstrates the effectiveness of each interpolation method as solution quality degrades over time.

The dissipative behavior of the linear approaches is illustrated in Fig. 36. Both bilinear and steering kernel regression (SKR) approaches preserve extrema and are therefore stable albeit less accurate. The SKR solution evolves in the same manner as the linear solution, but it initially maintains a slope 50% closer to the exact discrete

Table 7: Initial and final convergence rates for different interpolation methods applied to the two-dimensional step function described by Eqn. 63.

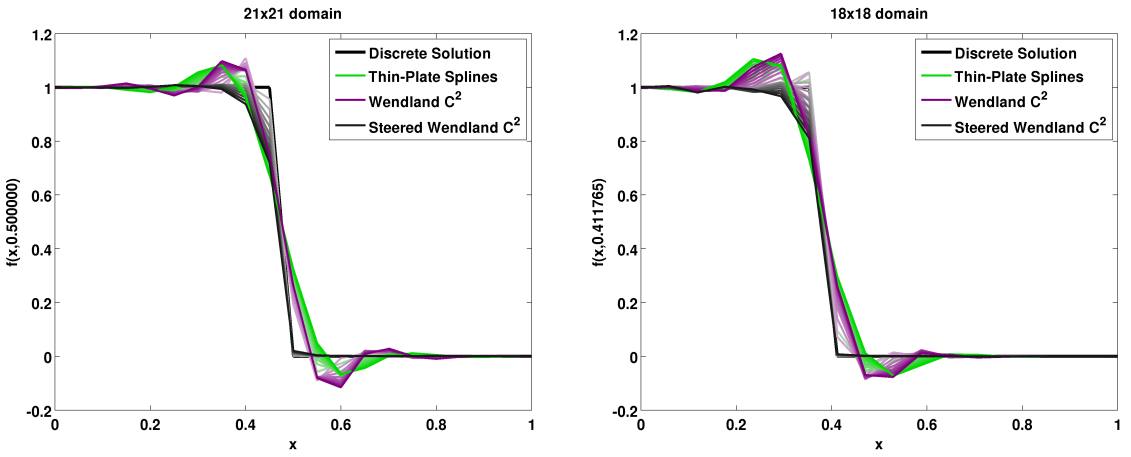
Interpolation Method	Initial Slope	Final Slope
Bilinear	0.56	0.0
Steering Kernel Regression	1.83	divergent
Thin-Plate Spline	0.59	0.0
Wendland C^2	0.56	0.0
Steered Wendland C^2	1.86	0.47



(a) Fine grid

(b) Coarse grid

Figure 36: Result of repeated interpolations between fine and coarse grids using linear methods. Evolution of transient solution is indicated by curves changing from light to dark color, and the discrete input solution (described by Eqn. 63) has been plotted for reference.



(a) Fine grid (b) Coarse grid

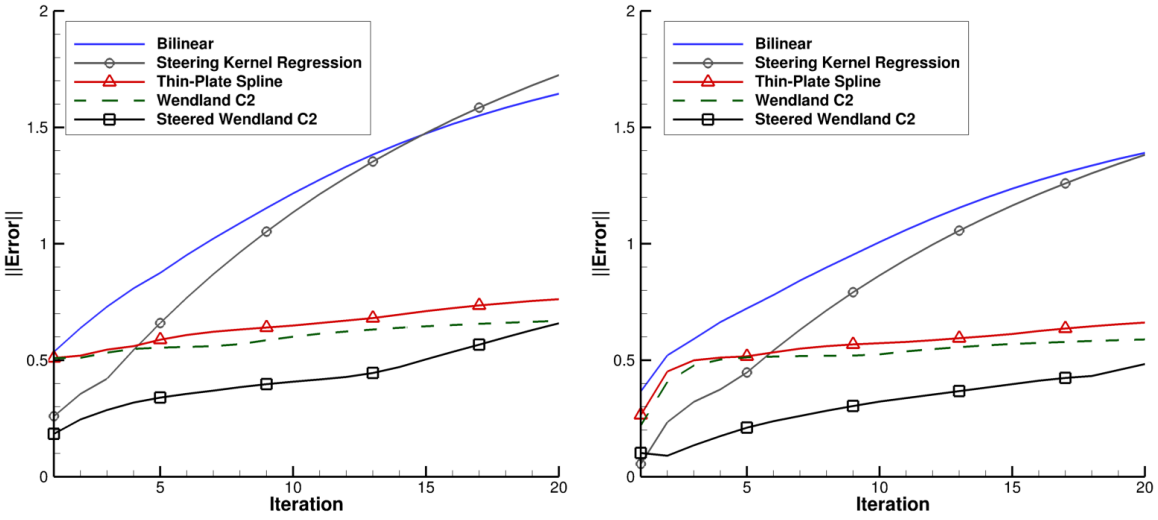
Figure 37: Result of repeated interpolations between fine and coarse grids using kernel interpolation methods. Evolution of transient solution is indicated by curves changing from light to dark color, and the discrete input solution (described by Eqn. 63) has been plotted for reference.

solution than bilinear interpolation. In comparison, the RBF-based methods maintain the slope at the discontinuity over many repeated interpolations (Fig. 37). However, the interpolated solution becomes unstable with Gibbs-like over- and under-shoot errors, increasing over time by nearly 20% after 25 iterations.

These findings are reflected in Fig. 38, which illustrates that the steered W2 approach has approximately half the error of the other RBF approaches in the short-term. The rate of error growth is relatively low at the onset because of solution preservation at the discontinuity, and because the dispersive errors have yet to significantly accumulate (Fig. 37). In comparison, the linear errors grow steadily from the onset as a result of dissipative error (Fig. 36).

In the asymptotic range, the errors on the coarse and fine grids are indistinguishable (Fig. 39) since the solutions on both grids have sufficiently degraded to the point where fine-to-coarse interpolation does not have any improved accuracy over coarse-to-fine interpolation because both data transfers start from a comparably poor solution. After $\sim \mathcal{O}(100)$ iterations, the dispersive over- and under-shoot errors begin

to accumulate more quickly and the solution becomes unstable. The instability is due to the ability of high-order approaches to reproduce solution features, regardless of how erroneous they may be (Fig. 37). While the steered W2 approach minimizes error initially, it still destabilizes at the same time as its baseline W2 function. In contrast, the TPS maintained stability for a longer period with errors not surpassing the linear approaches until $\sim \mathcal{O}(10^3)$ iterations.



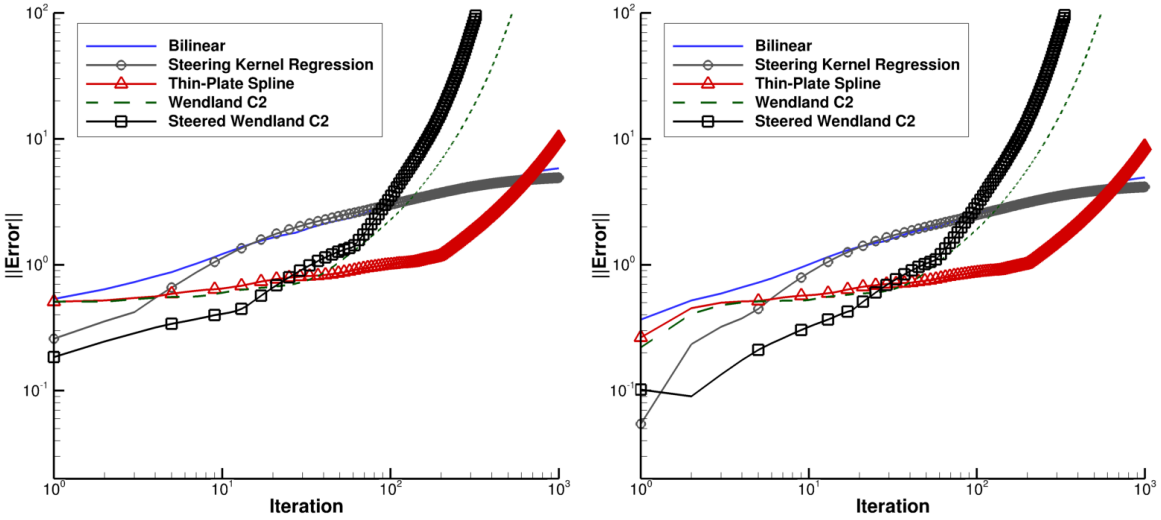
(a) Fine grid error

(b) Coarse grid

Figure 38: Initial evolution in root-mean-squared error from repeated interpolations between fine and coarse grids, using various linear and kernel interpolation methods.

4.5 Error and Conservation Test Cases

Following the work of Alauzet et al. [24], numerical experiments were conducted on unstructured grids for two different analytical test functions shown in Figs. 41 and 43. Two unique, similarly sized grids have been generated, and successively refined in a uniform manner. One mesh was created using the Delaunay-triangulation-based MATLAB initmesh and refinemesh codes, resulting in the mesh shown in Fig. 40a. A separate mesh was generated using the Mississippi State University Solidmesh software which is based on an advancing front algorithm, resulting in the mesh shown in Fig. 40b. In both cases, the input cell spacing was varied until a comparable



(a) Fine grid error

(b) Coarse grid

Figure 39: Asymptotic root-mean-squared errors resulting from repeated interpolations between fine and coarse grids, using various linear and kernel interpolation methods.

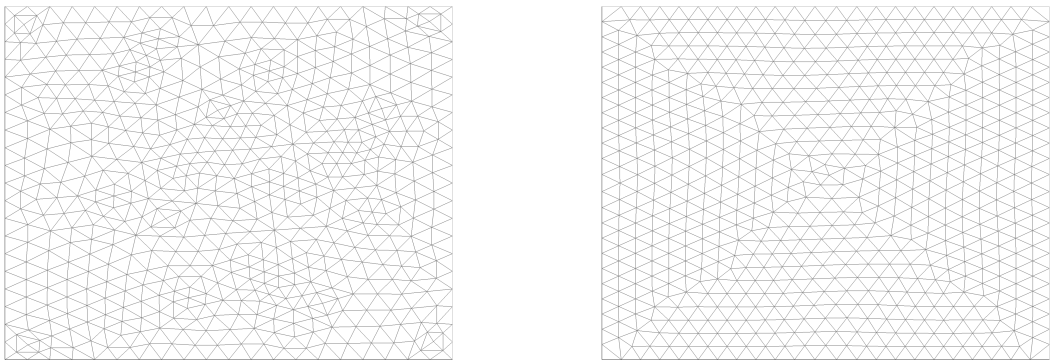
number of points was obtained, after which the grids were uniformly refined four times. Coincident points between meshes only exist at isolated locations (typically less than 10). Table 8 shows a summary of the final generated mesh sizes in comparison with the grids used in the Alauzet study [24]. The presentation of results follows a brief discussion on theoretical error estimates.

Table 8: Sizes of different meshes used in Alauzet numerical experiment.

Step	Alauzet $\mathcal{H}^1$	Alauzet $\mathcal{H}^2$	"A" (Delaunay)	"B" (Advancing Front)
1	631	611	625	639
2	244	2366	2417	2469
3	961	9311	9505	10139
4	38165	36941	37697	40244
5	152041	147161	150145	160343

In the present work, errors have been quantified by the L^2-norm in the same manner as Ref. 24:

$$\epsilon = \sqrt{\frac{\int_{\mathcal{H}}(u-\tilde{u})^2}{\int_{\mathcal{H}}(u^2)}} \tag{65}$$



(a) Delaunay ($N = 625$) (b) Advancing Front ($N = 639$)

Figure 40: Isotropic meshes (with number of nodes, N) generated for comparison with the Alauzet numerical experiments [24].

where u and $\tilde{u}$ are the exact and interpolated values on mesh $\mathcal{H}$, respectively. To evaluate the conservation properties of each interpolation scheme, the mass has been integrated on the source grid with the exact solution and on the target grid with the interpolated solution. The conservation error is taken to be the absolute difference between the two calculated values.

4.5.1 Theoretical Error Estimates

The error of a linear interpolation scheme is second-order, i.e., the errors are on the order of $\mathcal{O}(h^2)$ where h is the grid spacing. The grid spacing is in turn inversely proportional to the square root of the number of grid points on an isotropic grid ($h \sim N^{-1/2}$). Following the work of Ref. 24, errors have been plotted as a function of grid size, therefore theoretical convergence should be evaluated by comparing to a slope of -1.0.

While the theoretical error for linear interpolation is well known, the error associated with kernel function interpolation techniques requires a more sophisticated

analysis. The seminal work of Wu and Schaback [122] applied Lagrangian basis functions (also known as "cardinal" bases) to establish error bounds for radial basis function interpolation. Errors were further quantified in subsequent works [123, 124, 66]. These estimates are typically described in terms of a point density or *fill distance*, h:

$$h_{X,\Omega} = \sup_{x \in \Omega} \min_{1 \leq j \leq N} ||x - x_j||_2. \tag{66}$$

With a set of given data sites X from domain Ω, the fill distance is a measure of grid spacing in any dimensional space and can be qualitatively described as the radius of the largest generalized sphere that can be formed within a set of points without enclosing any other points. For scattered data (or anisotropic meshes), it is a more accurate measure of the characteristic local spacing between a set of points and thus a more accurate predictor of interpolation error. For meshes with relatively uniform cell sizes, the fill distance and the grid point spacing are approximately equal. Therefore within this section, the fill distance (h) and the grid spacing will be assumed to be the same quantity.

In the case of globally supported functions, different error bounds may be obtained depending on assumptions about the function space. Theoretical bounds have been presented in Table 9 but more optimal error bounds may exist and remain an area research [66]. While radial basis functions are generally unbounded, Lagrange functions are bounded and have known decay properties that are established by Fourier transformation [66]. This knowledge spurred initial analyses of the convergence of RBF interpolants as $h \to 0$. However, this approach assumes an infinite grid of data sites (source points) at integer multiples of h, which differs significantly from a typical scattered data application of RBF interpolation. The resulting error bound for thin-plate spline in n dimensions is:

$$||s_{f,X}(x) - f(x)||_\infty = \mathcal{O}(h^{n+2}), \qquad x \in \mathbb{R}^n. \tag{67}$$

For the scattered data case, a different analysis is required because it must account

for points x sourced from a now bounded domain Ω, and a lack of periodicity in the data. This convergence analysis applies variational theory and bounds the interpolant errors with a power function [66]. The new error bound is then:

$$||s_{f,X}(x) - f(x)||_{\infty,\Omega} = \mathcal{O}(h^{2k-\frac{n}{2}}), \qquad x \in \Omega \subseteq \mathbb{R}^n, \tag{68}$$

for which the sampled solution $f : \mathbb{R}^n \to \mathbb{R}$ should have its kth total degree distributional partial derivative also in the same linear function space for optimal convergence. With the set of polyharmonic spline basis functions defined more generally as

$$\phi(r) = \begin{cases} r^{2k-n} \log r, & 2k - n \text{ an even integer} \\ r^{2k-n}, & 2k - n \text{ an odd integer} \end{cases} \tag{69}$$

the standard definition of a thin-plate spline ($\phi(r) = r^2 \log r$) yields $k = \dfrac{n+2}{2}$. Substitution into Eqn. 68 yields an error bound of $\mathcal{O}(h^{n+2-n/2})$, which is a factor of $h^{-n/2}$ off from the optimal, infinite uniform grid case [66]. A summary of all error bounds are presented in Table 9. For the convergence estimates with assumed smooth functions, the theoretical error is on the order of $\mathcal{O}(h^{\mu})$ only if the function f is μ times continuously differentiable, and the $\lceil \mu \rceil$ and $(\lceil \mu \rceil - 1)$ total derivatives are uniformly bounded. The theoretical convergence rates are higher for more uniform data and in higher dimensions. Since aerodynamic solutions are in general not smooth everywhere, the optimal convergence rates are not expected. Observed convergence rates for all test cases in this thesis were at minimum comparable to standard second-order linear approaches ($\mathcal{O}(h^2)$ with smooth functions).

For the Wendland family of compactly supported radial basis functions, theoretical error bounds are readily available [47]. Theoretical convergence estimates indicate that the Wendland C^2 function is slower than thin-plate splines for smooth data but faster than the upper bound of thin-plate splines on scattered data. Experimentally observed rates of convergence by Costin and Allen [107] were higher, with fourth- to

Table 9: Theoretical rate of convergence of RBF interpolation errors in terms of fill distance h, for source points in n dimensions. Convergence orders for smooth data are achieved only for functions that are smooth up to the same order.

Basis Function	$n = 2$	$n = 3$	Reference
Thin-Plate Splines, smooth uniform data	$\mathcal{O}(h^4)$	$\mathcal{O}(h^5)$	[66]
Thin-Plate Splines, smooth scattered data	$\mathcal{O}(h^3)$	$\mathcal{O}(h^{3.5})$	[66]
Thin-Plate Splines, scattered data	$\mathcal{O}(h)$	$\mathcal{O}(h)$	[122]
Wendland C^2, scattered data	$\mathcal{O}(h^{2.5})$	$\mathcal{O}(h^{2.5})$	[47]

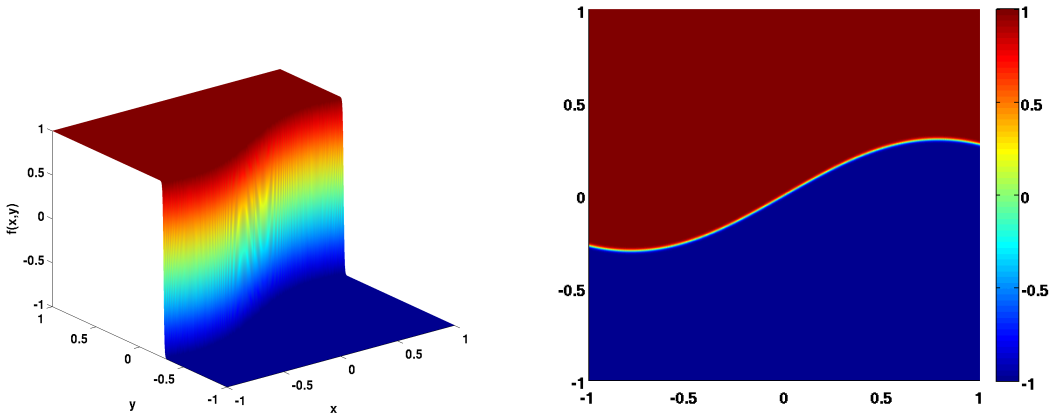


Figure 41: Continuous sinusoidal shock function defined in Eqn. 70.

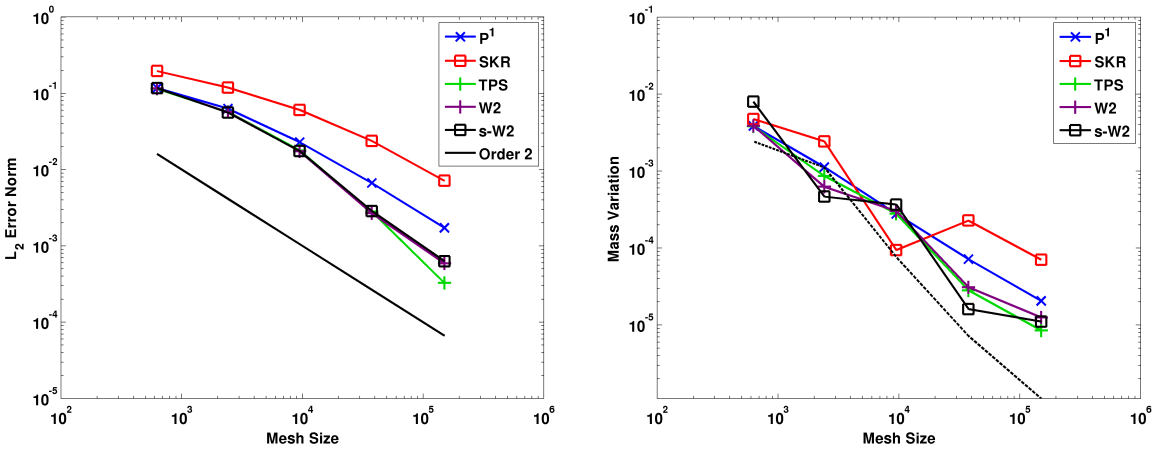
sixth-order accuracy in one and two dimensions and third-order accuracy or higher in three dimensions. A maximum rate of error convergence on the order of $\mathcal{O}(h^5)$ was observed in three-dimensions but required a neighborhood of 60 points.

4.5.2 Continuous Sinusoidal Shock

A discrete representation of a discontinuity is modeled as a continuous sinusoidal shock in Eqn. 70, and shown in Fig. 42.

$$f(x, y) = \tanh(100(y + 0.3\sin(-2x))) \tag{70}$$

In terms of L^2-norm, the errors are on the same order for all methods up to a mesh size on the order of 10^4 nodes. For larger meshes, the strength of the RBF approach becomes more apparent as errors achieve a greater rate of convergence in terms of



(a) Error (b) Mass Conservation Error

Figure 42: Continuous sinusoidal shock interpolation experiment from Ref. 24. Results are shown for linear (P1), steering kernel regression (SKR), thin-plate splines RBF (TPS), Wendland C^2 RBF (W2), and steered RBF (s-W2) methods. The dashed line in (b) is the error in the numerically calculated exact mass.

increasing mesh size and consequently decreasing fill distance. The greatest rate of convergence is achieved with the classic thin-plate splines function compared to the linear method. In the linear case, the asymptotic rate of convergence is approximately linear, with a slope of -0.98; in comparison, the TPS interpolation achieves the highest rate of convergence with a final slope of -1.56. Perhaps more importantly, the error is consistently less than or equal to the linear error with the final error being an order of magnitude lower than the standard linear approach.

Using compactly-supported RBF functions also sees a reduction in error by almost an order of magnitude, even though the final rate of convergence is still comparable to the linear case with a slope of -1.10 (Fig. 42a). The steered RBF approach was less successful. This may be at least partly due to the interpolation including only about 30% or less of the original source points. In comparison, the structured grid cases previously evaluated retained nearly 70% of the set of source points (discussed in Section 4.4).

Since a regression approach is typically applied to noisy data, it approximates

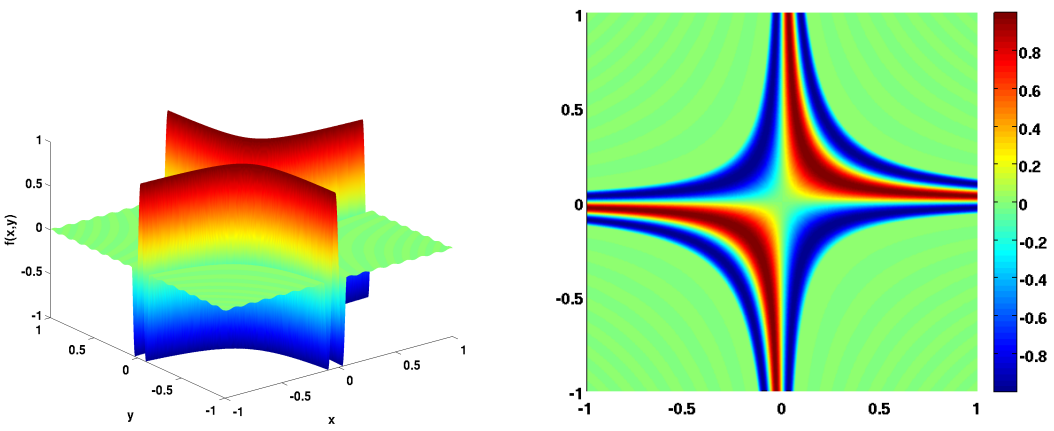
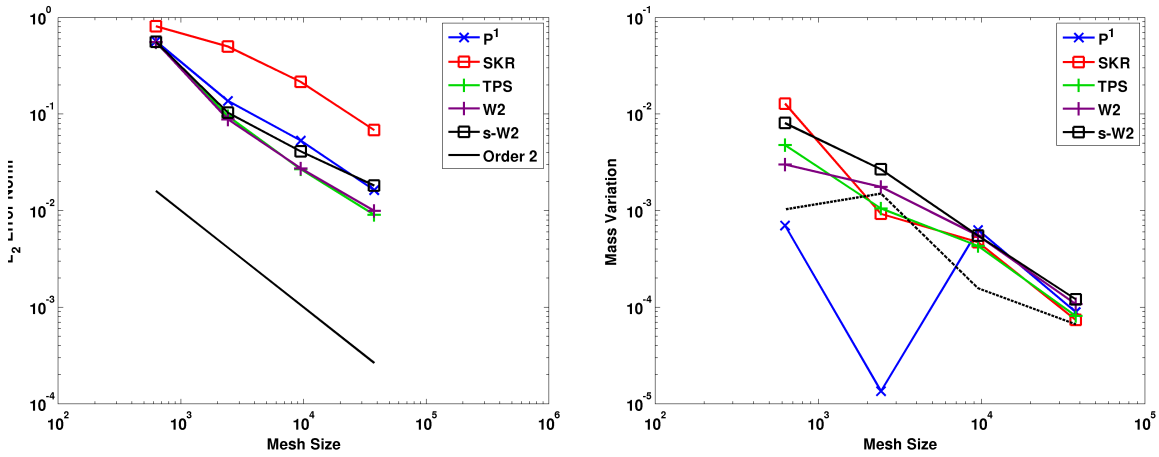


Figure 43: Multi-scales smooth function defined in Eqn. 71.

the solution field and does not guarantee exact reproduction of input points. The ineffectiveness of the steering kernel regression for this case meant that the SRBF performance was also degraded. While the steered approach in general had slightly higher errors compared to the standard W2, in all cases the errors were at worst comparable to the linear mapping Fig. 42a.

The effects of different interpolation techniques on conservation errors were mixed in general. With the exception of the coarsest mesh level, the SRBF approach gave the lowest conservation error with an order of magnitude reduction at best (Fig. 42b). All of the RBF approaches were convergent, in that the error with increasing mesh size was less than or equal to the error on the coarser mesh. Results from Alauzet et al. [24] were not exactly reproduced due to differences in geometry and mass calculation method. The non-monotonic behavior of the conservation error curves can be related to the error in the numerical estimate of the mass (shown as a dashed curve) calculated based on exact rather than interpolated values. Results in Fig. 42b are therefore not reliable where the calculated mass conservation error is on the same order as the error in the calculated mass, i.e., for mesh sizes less than 10^4.



(a) Error (b) Mass conservation

Figure 44: Multi-scale smooth function interpolation experiment from Ref. 24. Results are shown for linear (P1), steering kernel regression (SKR), thin-plate splines RBF (TPS), Wendland C^2 RBF (W2), and steered RBF (s-W2) methods. The dashed line in (b) is the error in the numerically calculated exact mass.

4.5.3 Smooth Multi-Scale Function

A smooth, multi-scale function is described by Eqn. 71, and depicted in Fig. 44. The ability to reliably reproduce functions with multiple inherent scales is important for the interpolation of turbulent quantities, which are characterized by highly irregular fluctuations associated with the interactions between fluid structures at different spatial scales.

$$f(x,y) = \begin{cases} 0.01\sin(50xy) & \text{if } xy \leq -\pi/50 \\ \sin(50xy) & \text{if } -\pi/50 < xy \leq 2\pi/50 \\ 0.01\sin(50xy) & \text{if } 2\pi/50 < xy \end{cases} \tag{71}$$

This test case is more challenging and all approaches exhibit the same limiting behavior in terms of conservation error. In terms of interpolation accuracy, the standard RBF approaches still consistently outperform the linear approach and the SRBF has errors approximately comparable to the linear approach at the finest grid level (Fig. 44a). In terms of mass conservation (Fig. 44b), the variation in mass for the

linear approach is because the error in integrating the exact mass of the input function is of the same order as the interpolated mass estimate for first-order integration. None of the advanced approaches (except for SKR) perform any worse than the linear approach, which again also degrades the effectiveness of the SRBF method. However, there are a number of reasons why the SKR approach is again ineffective. The established baseline parameters such as steering neighborhood size may be inadequate for unstructured meshes. In addition, the first-order gradient estimate is unlikely to adequately detect the high wavenumber features in the function (previously illustrated in Fig. 43).

4.6 Interpolation of Turbulent Flow Fields

The turbulent flow field from a three-dimensional, unstructured grid URANS solution of a frigate in headwind flow (Fig. 45) has been interpolated onto intermediate Cartesian meshes with six uniform spacing levels of $\Delta s = 0.2$, 0.1,...0.00625. Solutions from the Cartesian mesh are then interpolated back onto the original unstructured mesh. The coarse spacing ($\Delta s = 0.2$) is comparable to a mesh spacing that might be used by a Cartesian wake solver in a hybrid computational methodology. The solution field (in terms of streamwise velocity) is illustrated on the colored mesh in Fig. 45, with a Cartesian test grid (at $\Delta s = 0.05$) in overlay. Grids both finer and coarser than depicted have been evaluated using linear techniques including bilinear mapping and steering kernel regression), as well as RBF techniques employing thin-plate splines (TPS), Wendland C^2 (W2), and steered Wendland C^2 functions.

For the cases in which a steering strategy has been applied (steering kernel regression and steered W2 interpolation), steering parameters were calculated using estimated gradients (first-order on the unstructured grid, and second-order on the interior and boundaries of the Cartesian grid). The first-order approach has been used to facilitate the demonstration and is expected to be sufficient given the relatively

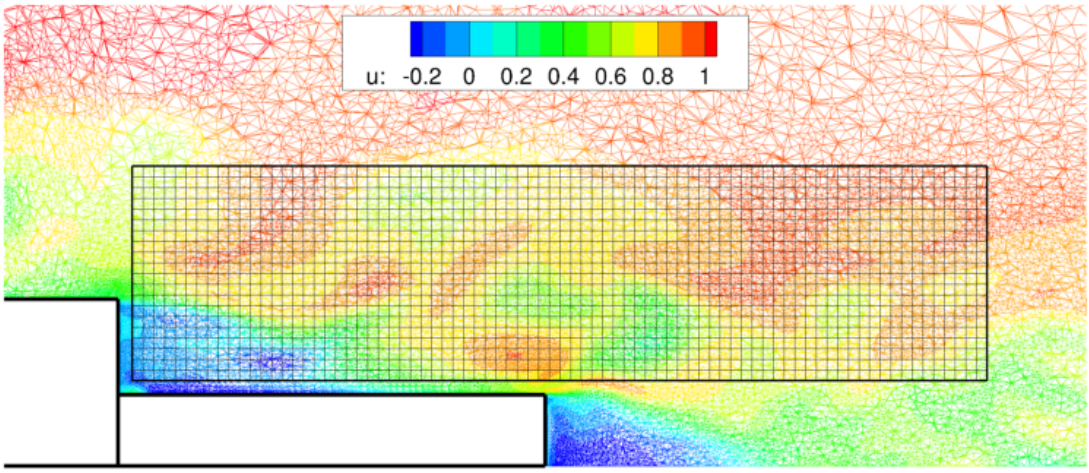


Figure 45: Naval ship airwake contours of streamwise velocity. The outlined region indicates the output Cartesian grid ($\Delta s = 0.05$) used for the interpolation demonstration.

fine input mesh, especially in the regions aft of the hangar and above the flight deck (Fig. 45). Calculated steering kernels (based on the first-order gradient calculations) are illustrated in Fig. 46. Instead of plotting elliptical kernel footprints as before, vectors along the semimajor axis of each support region are plotted, with the vector length corresponding to the magnitude of the elongation parameter.

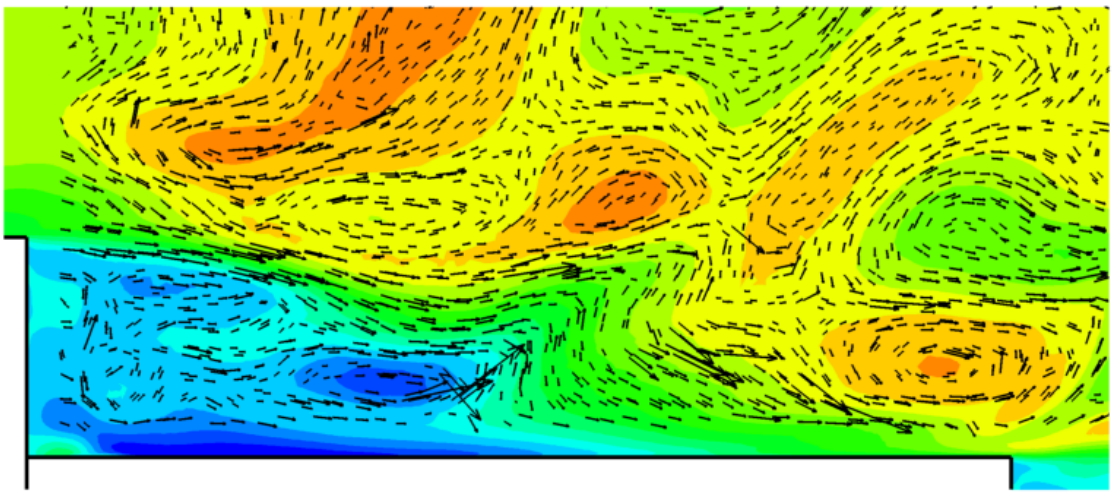


Figure 46: Anisotropy in streamwise velocity of a naval ship wake detected by steering strategy. Vector length corresponds to elongation of the kernel footprint.

The evolution of the error in all solution quantities is presented in Fig. 47. In all cases, the steering kernel regression approach has up to half an order of magnitude higher error than all other interpolation techniques. This reaffirms the results from Section 4.5, which demonstrated that on unstructured grids the regression approach can have errors comparable to linear interpolation or higher. In addition, the results from Section 4.3 revealed that for relatively smooth, low-gradient fields, the regression approach is not consistently more effective than linear approaches.

The next highest errors after the steering kernel regression were consistently the linear interpolation results. At coarser mesh levels, errors were comparable across all approaches, but as the Cartesian mesh was refined, error reductions of up to 50% compared to linear interpolation were observed in all mean flow and turbulence solution quantities when applying RBF techniques (Fig. 47). For the velocity variables, the steered RBF technique shows 5–10% additional error reduction (Fig. 47a and b) over the baseline RBF approaches at moderate grid refinement (i.e., $\Delta s = 0.05$ as depicted in Fig. 45). However, at the finest grid levels tested, the error reduction was 50% for RBF and 40% for the steered RBF techniques. Errors in eddy viscosity are similar at the finest output grid level (Fig. 47d). For pressure and turbulent kinetic energy (Fig. 47c and e), similar error reductions (on the order of 50%) were observed, with the W2 approaches (steered and unsteered) having approximately 5% less error than TPS.

While the steered RBF results showed modest improvements at coarser grid levels, its effectiveness tended to decrease with increasing mesh refinement. This is because the number of points near which large gradients exist tends to stay constant, while the number of smooth field points is increased. As was demonstrated in Section 4.3, steering-based approaches are most effective in the presence of large, well-defined gradients. Therefore as the mesh increases in size, the number of points at which the steering is truly effective is reduced and the ability of the steered RBF technique to

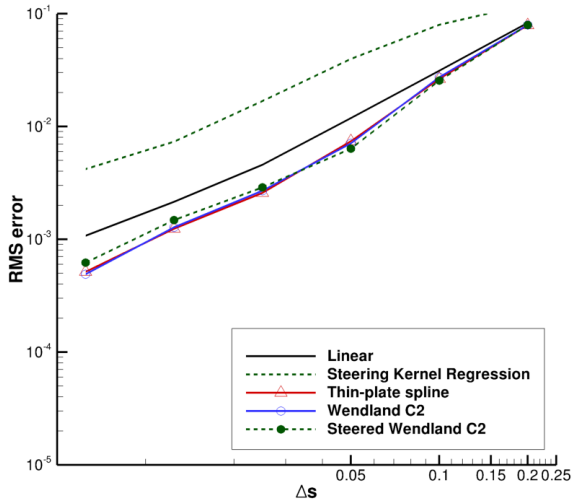
reduce overall error is also diminished.

4.6.1 Gradients of the Interpolated Solution

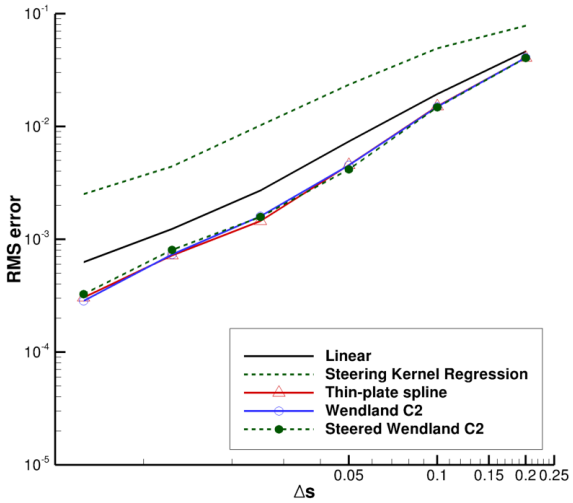
Gradients of all solution variables are calculated from the interpolated flow field on the original unstructured mesh. As with the estimation of steering parameters, gradients are calculated with a first-order approximation for all the approaches. Even though the RBF techniques provide a differentiable mapping of the local solution field, gradients for all interpolated solutions as well as the reference gradient field have been computed in identical fashion to permit comparisons. Figures 48 and 49 illustrate the errors in the interpolated velocity gradient field. Overall, the linear and unsteered RBF approaches in general produced very similar errors. The steering kernel regression, while being data-adaptive, provides inconsistent improvements since the regression approach does not guarantee exact reproduction of input data. However, the steered Wendland C^2 interpolation demonstrated consistent improvement over the linear gradient estimates. In addition, at the finest Cartesian grid levels, this approach produced the lowest gradient errors for all solution quantities (mean flow and turbulent), with up to 81% reduction in the streamwise direction and up to an order of magnitude reduction in the normal direction.

For the gradients in the streamwise direction (Fig. 48), the errors in the linear and unsteered RBF solutions are nearly identical, for both mean flow and turbulence quantities. Exceptions include the pressure gradient, where for small Δs the unsteered RBF solutions (with thin-plate spline or Wendland C^2 interpolation) have 9% less error. In comparison, the steered Wendland C^2 approach reduces gradient error by over 50% in the same variable. Similarly, for the eddy viscosity solution, the thin-plate splines and Wendland C^2 radial basis function reduce error by 17% and 27% respectively, while the steered approach is able to reduce error by 50%.

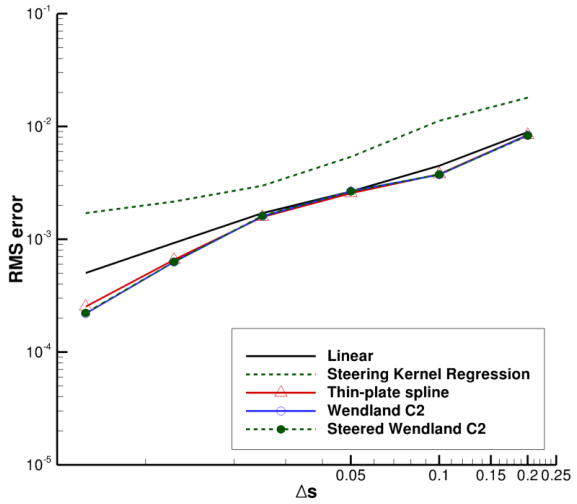
For the gradients in the direction normal to the flow (Fig. 49), the normal velocity



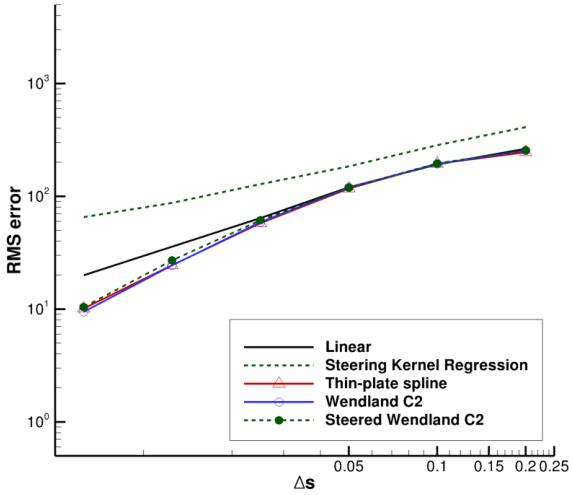
(a) Streamwise velocity



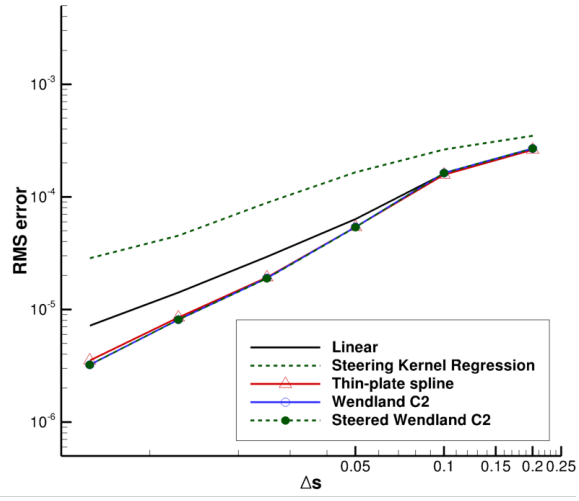
(b) Normal velocity



(c) Pressure

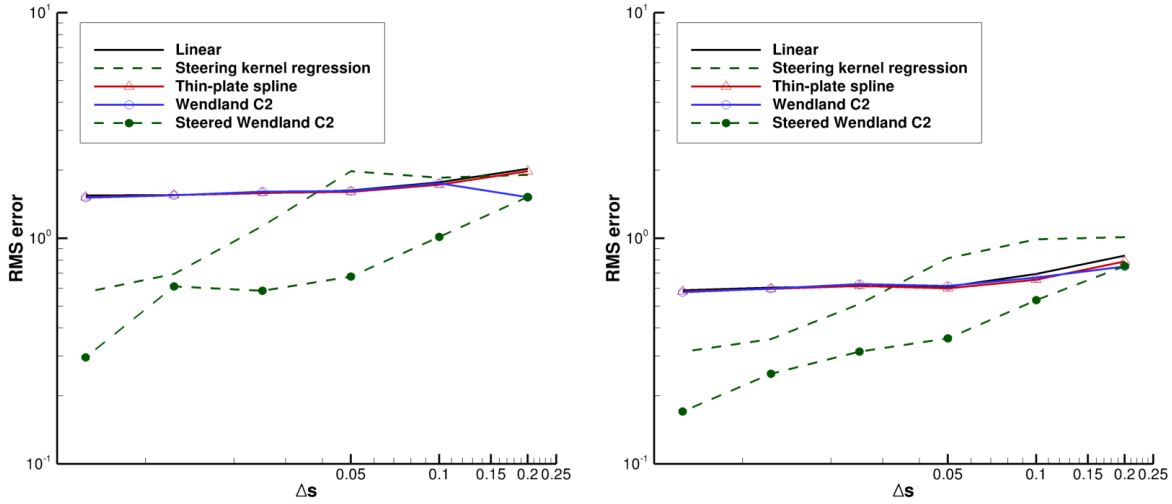


(d) Eddy viscosity



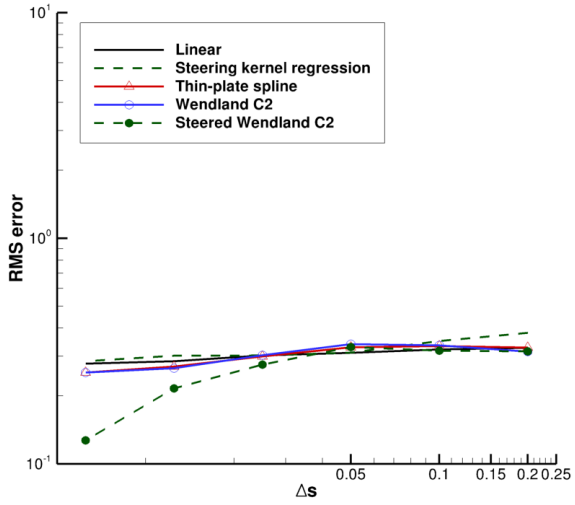
(e) Turbulent kinetic energy

Figure 47: Root-mean-squared errors for an interpolated turbulent solution field in the wake of a naval ship model.

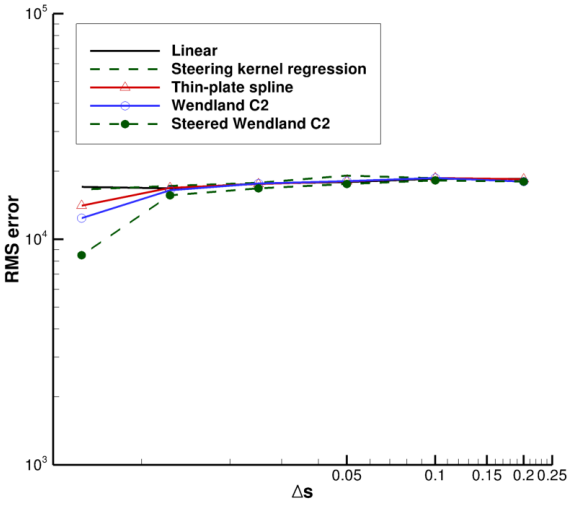


(a) Streamwise velocity

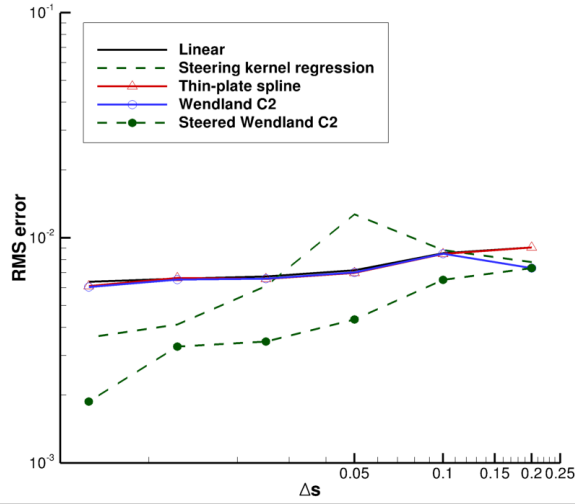
(b) Normal velocity



(c) Pressure

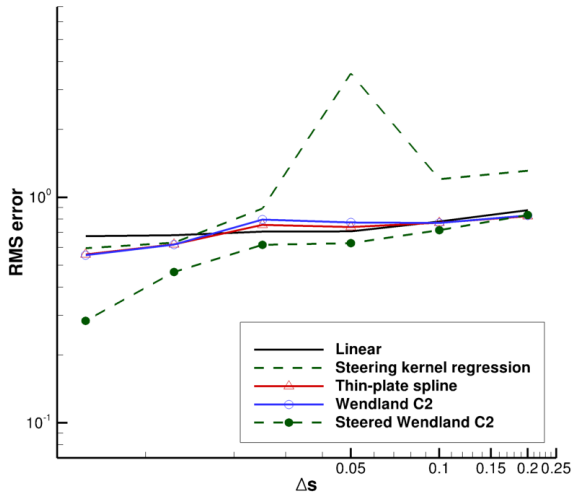


(d) Eddy viscosity

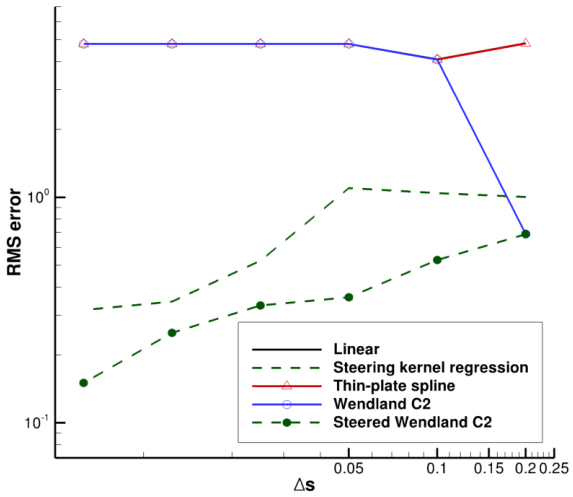


(e) Turbulent kinetic energy

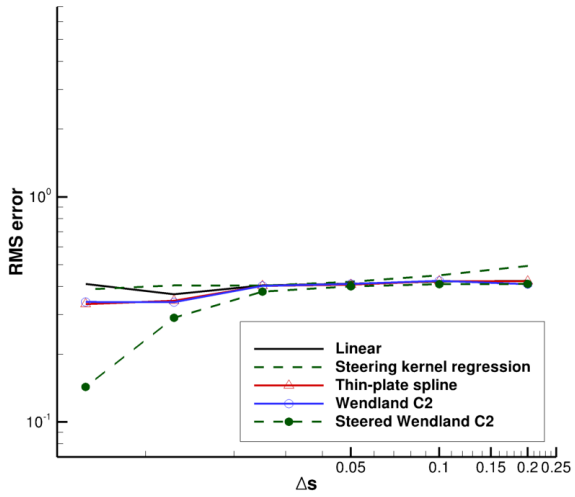
Figure 48: Root-mean-squared errors for solution gradients in the streamwise direction.



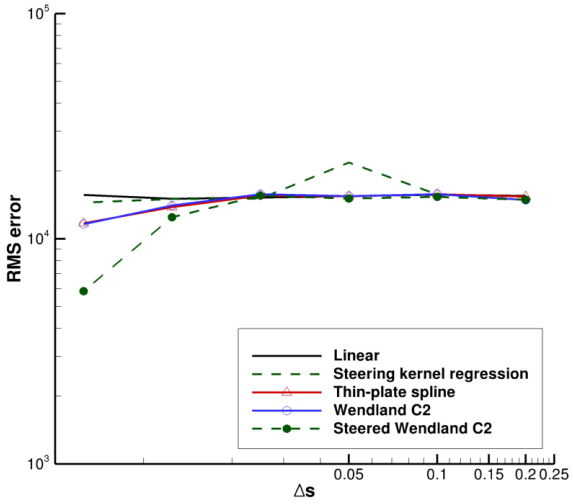
(a) Streamwise velocity



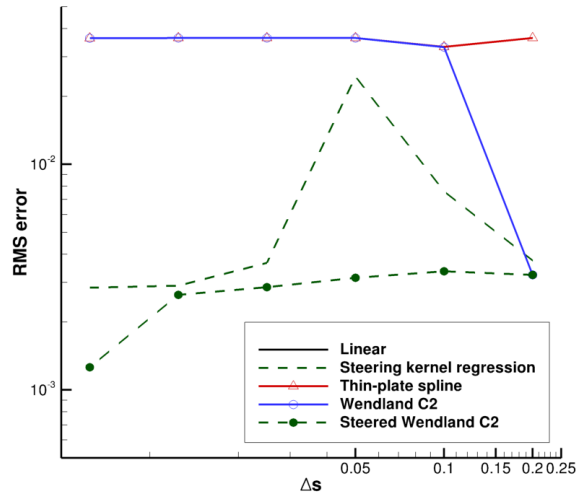
(b) Normal velocity



(c) Pressure



(d) Eddy viscosity



(e) Turbulent kinetic energy

Figure 49: Root-mean-squared errors for solution gradients in the normal direction.

and turbulent kinetic energy solutions tend to be undifferentiated between the linear and unsteered RBF approaches. For these solutions interpolated from the coarsest Cartesian mesh ($\Delta s = 0.2$), the thin-plate splines and linearly interpolated errors are virtually identical. In comparison, order of magnitude reductions in error for these quantities were observed when applying the steered Wendland C^2 approach. For the other solution quantities (streamwise velocity, pressure, and eddy viscosity), the unsteered radial basis function approaches had 17–26% less error. In comparison, the steered Wendland C^2 approach achieved 58–65% reductions in error.

4.6.2 Convergence

The rates of convergence when applying various linear and RBF interpolation techniques have been calculated based on the RMS error (depicted in Fig. 47) in the asymptotic range ($\Delta s \leq 0.025$). These rates are summarized in Table 10. Convergence of flow field gradients have not been considered as those errors include both interpolation errors and gradient estimation errors. Sampled variables include streamwise velocity (u), normal velocity (w), pressure (p), eddy viscosity (μ_t), and turbulent kinetic energy (k). When applying linear interpolation, linear rates of convergence were observed in velocities and turbulent kinetic energy. Rates slower than linear convergence were observed for interpolated pressure and eddy viscosity. The steering kernel regression approach also demonstrated linear convergence in interpolated velocities, but exhibited only nearly-linear convergence in k. Moreover, the rates of convergence in pressure and μ_t are further reduced.

In contrast, all radial basis function approaches demonstrated higher than first-order convergence, at a 10–43% higher rate than linear (Table 10). Overall, the Wendland C^2 function was most effective at both reducing error and achieving the highest convergence rate in all interpolated variables. The thin-plate splines and

Table 10: Convergence of interpolation errors when applying linear and radial basis function interpolation in a turbulent ship airwake.

Method	u	w	p	μ_t	k
Linear Interpolation	1.05	0.96	0.80	0.84	1.02
Steering Kernel Regression	1.00	1.01	0.40	0.48	0.82
Thin-Plate Spline	1.16	1.12	1.32	1.25	1.22
Wendland C^2	1.22	1.24	1.43	1.31	1.26
Steered Wendland C^2	1.10	1.13	1.43	1.27	1.26

steered Wendland C^2 methods have convergence rates within 10% of the best Wendland C^2 method for mean flow variables (u, w, and p) and were within 5% of Wendland C^2 for the turbulence variables (μ_t and k).

CHAPTER V

OVERSET TEST CASES

The advanced mapping methods discussed in Chapter 3 have been successfully applied to interpolate scattered, unstructured data for a variety of test cases. This chapter will assess the effectiveness of radial basis function (RBF) techniques within the context of actual overset URANS simulations. Recent development of advanced mapping schemes has focused on high-order polynomial interpolation for structured grids [18, 44]. The current work develops scattered data approximation techniques using kernel function interpolation to transfer overset data on unstructured grids. Three problems have been studied for validation: a two-dimensional inviscid convecting vortex, a two-dimensional shock tube, and a three-dimensional turbulent ship airwake. The first problem exemplifies a smoothly varying flow field, while the shock tube shipwake problems assess the effectiveness of the advanced techniques in reproducing discontinuous fields. The convecting vortex problem is revisited to evaluate the effectiveness of scattered data techniques in mitigating configurations with orphan points.

5.1 Description of Code Modifications

Key modifications to the Suggar++ and DiRTlib connectivity software (described in Section 2.4.3) to permit kernel-based interpolation are outlined below:

1. Identification of local point clouds (within Suggar++): A search for a cloud of donor points (also known as source points) is performed for each receptor (target). For a node-centered analysis as performed by FUN3D, the standard search returns a stencil of donor points from one (or more) grids in proximity to the

point of interest. Within the Suggar++ paradigm, if the interpolant depends on more than one grid, the donors are considered "fragmented." The interpolation problem has been defined in a manner that allows fragmented donors and permits efficient parallel operation by re-ordering the columns of the inverted interpolation matrix defined in Eqn. 28.

For the purposes of kernel-based interpolation, there is no appreciable reduction in error when utilizing a linear interpolation stencil based only on the nodes of the enclosing donor cell (discussed in Section 3.3.1), so additional points should be added to the interpolation cloud. The Suggar++ donor search is thus modified to return not only the nodes encompassing the receptor node, but neighbors of these nodes as well. Connectivity information has been used for convenience to avoid having to perform additional nearest neighbor searches.

2. Interpolation Problem Setup (in Suggar++): Once the locations of the donors and receptors are known, the left-hand side interpolation matrix (Eqn. 28) can be calculated. If the simulation does not have moving bodies, then the connectivity (in terms of associated donor and receptor points) is known a priori. At the cost of increased memory usage, the interpolation matrices are inverted once during the preprocessing stage, and at each time step/subiteration the interpolation coefficients are computed by multiplying the saved decomposed left-hand side matrix with a vector of current solution values. This produces an interpolant that has dependence both on spatial location and the local solution field.

3. Data Transfer (between Suggar++/DiRTlib): At each iteration, the flow solver reads into memory the domain connectivity information (DCI) for each receptor point so that DiRTlib can calculate fringe point values. Suggar++ has been

modified to write a modified DCI file that includes both the decomposed left-side matrix and a vector of pre-calculated RBF values (Eqn. 26 in matrix form) to multiply with interpolation coefficients.

4. Interpolation Weight Update (in DiRTlib): An extra step is performed to calculate the interpolation weights using pre-calculated interpolation matrices before normal library routines multiply the weights with updated donor values at each iteration.

The RBF interpolation feature has been fully integrated into the Suggar++/DiRTlib framework and is enabled with a single option in the Suggar++ input file.

5.2 Inviscid Convecting Vortex

A benchmark test case involving an inviscid, convecting vortex has been used for validation. This test case has been evaluated by several authors [45, 18, 44] for high-order overset interpolation, with errors and convergence properties carefully evaluated for Lagrangian and B-splines methods. The RBF approach presented here is easier to implement than an implicit Lagrangian method because a formal interpolation stencil is replaced with arbitrarily defined clouds of source points. This enables natural extension of the method to unstructured grids and grid configurations that may include orphan points. For this configuration, the thin-plate spline (TPS) and compactly-supported Wendland C^2 (W2) basis functions have been evaluated.

5.2.1 Experiment Description

A vortical disturbance was written in a form consistent with the non-dimensionalization scheme in FUN3D [125]:

$$\frac{u}{U_\infty} = 1 - \frac{C}{U_\infty R}\frac{y - y_c}{R}\exp\left(\frac{-r^2}{2}\right) \tag{72}$$

$$\frac{v}{U_\infty} = \frac{C}{U_\infty R}\frac{x - x_c}{R}\exp\left(\frac{-r^2}{2}\right) \tag{73}$$

$$\frac{p - p_{norm}}{\rho_\infty U_\infty^2} = 1 - \frac{C^2}{2U_\infty^2 R^2}\exp(-r^2) \tag{74}$$

$$r^2 = \frac{(x - x_c)^2 + (y - y_c)^2}{R^2} \tag{75}$$

for constant density $\rho = \rho_\infty$, free stream reference velocity U_∞, and $p_{norm} = p_\infty - \rho_\infty U_\infty^2$. The vortex is centered at (x_c, z_c) with a nominal core radius R of 1.0 and non-dimensional vortex strength $C/U_\infty R$ of 0.02, respectively. The pressure expression was obtained by integrating $\dfrac{\partial p}{\partial r} = \dfrac{\rho_\infty u_\theta^2}{r}$ [126].

The key difference in computational configuration between the Sherer and Scott results [18] and the present effort is that FUN3D is an unstructured overset grid methodology with second-order inviscid fluxes [51]. Sherer and Scott applied an overset scheme on structured grids that varied in spatial accuracy from fourth-order on the boundary up to sixth-order on the interior of the computational domain. To achieve the same baseline levels of accuracy, the "fine" mesh spacing of $\Delta = 0.2$ from the Sherer study was employed as the current baseline case, and an additional refined grid with $\Delta = 0.1$ was generated resulting in 21 points across the vortex core. The time step for the simulation is $\Delta t = 0.005$ to replicate the Sherer computational setup [18].

5.2.2 Grids

Cartesian background meshes with similar sizes to Ref. 18 were generated (Fig. 50a), in addition to unstructured, mixed-element patch grids (Fig. 50b). Unstructured grids

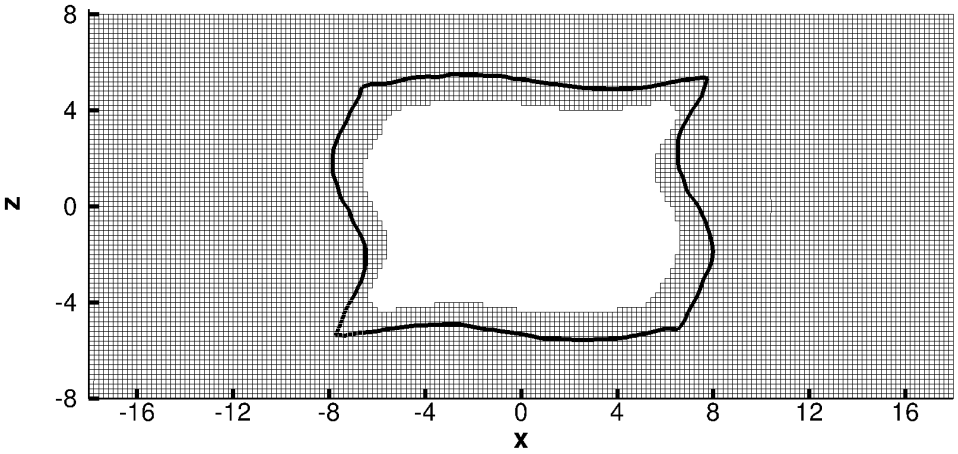
Table 11: Unstructured grids used in the vortex convection study

Grids	Grid Spacing (Δs)	Nodes	Fringes
Vortex, single grid	0.2	29,322	-
	0.1	116,242	-
Vortex, overset	0.2	36,620	1,826
	0.1	145,410	3,926

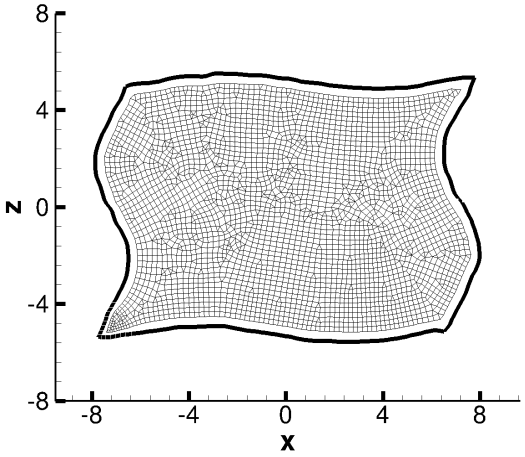
were used to model the patch because the irregular geometry can be more accurately represented by an unstructured topology. The composite overset mesh was formed by combining the meshes in Fig. 50 using the modified Suggar++ software described in Section 5.1. Resulting composite grids are described in Table 11 where Δs describes the length of an isotropic cell edge. An additional cutter grid (not shown) was used by Suggar++ to cut out the hole in the background; this is a reduced version of the patch grid and is not present during flow simulation. Two different grid sizes have been considered in this validation case, with the refined grid formed by multiplying the number of points in each direction. The resulting number of fringe points at which flow data are transferred is at least an order of magnitude smaller than the total number of grid nodes (Table 11), so the local increase in memory usage does not have a significant impact on overall simulation memory requirements.

5.2.3 Accuracy

Figure 51 illustrates typical results for a single grid compared with linear interpolation and with neighborhood-based RBF interpolation applying thin-plate spline (TPS) and the Wendland C^2 (W2) basis functions. The standard interpolation is an inverse isoparametric mapping (discussed in Section 3.1.3) with first-order elements, where interpolation from the background grid onto the irregular patch grid is equivalent to bilinear interpolation. In general, the largest reduction in error is achieved by refining



(a) Cartesian background grid with irregular cutout



(b) Irregular patch grid

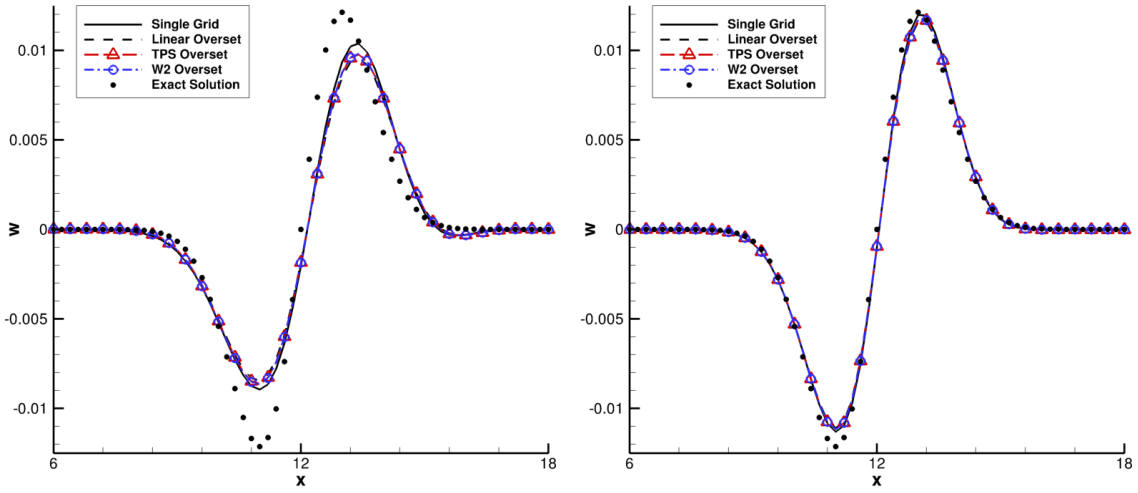
Figure 50: Overset grid systems for advanced overset investigations ($\Delta s = 0.2$); outlined region denotes the extent of the patch grid.

the mesh. For both mesh spacings, the most accurate solution is given by the non-overset, single-grid solution, as expected. However, in the close-up view of Fig. 51, the RBF solutions demonstrate improvements in accuracy over the linear approach. For this simple test case, the peak swirl velocity from the standard overset solution (which applies a linear mapping for interpolation) differs from the non-overset case by 8.0%; in comparison the TPS and W2 overset solutions only have 5.9% and 5.6% error, respectively.

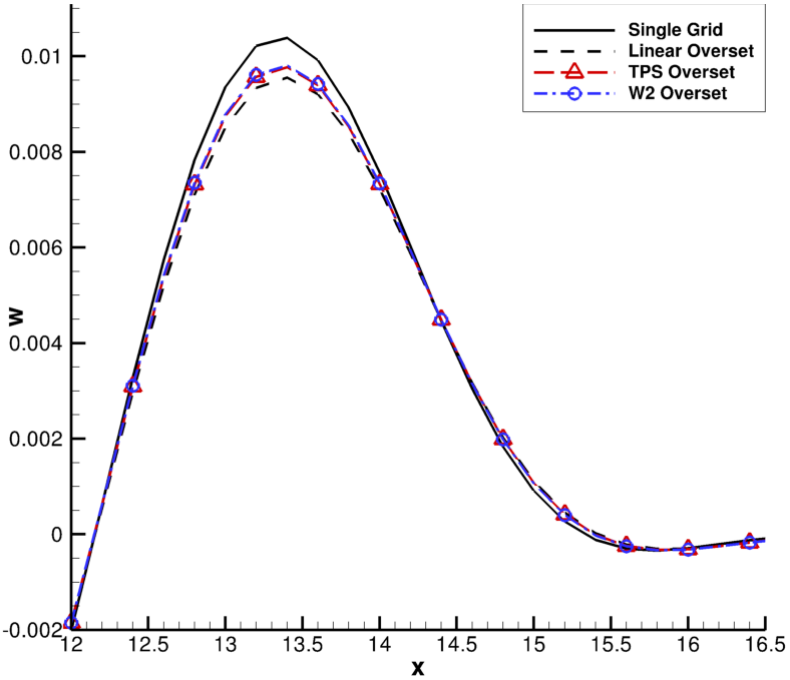
5.2.4 Cost

Tables 12 and 13 summarize typical timings for the preprocessing and fringe interpolation aspects of an overset URANS simulation. The initial preprocessing is performed with the modified Suggar++ code. Since this is a static simulation (i.e., there are no moving meshes), most of the overset cost is accrued in advance as the donor-receptor pairs do not change. For the current structured background grid configuration, the *local* case locates the cell that encloses the receptor node, and includes all eight adjacent hexahedral cell nodes as donors. This approach produces results of identical quality to the linear approaches, as demonstrated by the PDE example in Section 3.3.1. The *neighborhood* case selects a cloud of nearby points by also including the nodes to which the local eight nodes are connected. Since this is a two-dimensional simulation, the grid is only one cell deep (i.e., it is made up of two parallel planes of nodes), and the interpolation cloud includes up to 32 donor nodes.

In terms of preprocessing cost, the time required to set up the interpolation system scales with the number of nodes (Table 12). For example, uniform grid refinement from $\Delta = 0.2$ to 0.1 increases the number of points and the preprocessing time by a factor of 4 (on a two-dimensional mesh) for both local and neighborhood approaches. Between the two approaches, the number of donor points processed increases from 8 to 32 (by a factor of four), but the preprocessing time increases by a factor of



(a) $\Delta s = 0.2$ (b) $\Delta s = 0.1$



(c) $\Delta s = 0.2$, close up

Figure 51: Calculated swirl velocities with linear and RBF-based overset interpolation techniques at the end of the convecting vortex simulation.

five. The additional cost is associated with locating the additional source points and verifying that the donor list is unique.

During the CFD calculation, the computational cost of each approach scales similarly with increasing mesh refinement. The cost of the neighborhood technique is

an order of magnitude larger than the local approach including eight adjacent nodes (Table 13). There is no direct impact on the rate of convergence with the improvement from the neighborhood approach. This can be confirmed by the average number of subiterations per step required before the reduction in temporal error met a convergence criteria of one order of magnitude error reduction. On finer grids, the cost of interpolation becomes negligible in comparison to the total computational cost (Table 13). For example, the fraction of the total wall-clock time decreases by nearly 7% for the neighborhood approach despite an increase in the number of points by a factor of four.

Table 12: Measured wall-clock time (in seconds) for preprocessing the inviscid convecting vortex case.

Test Case		**Total processing time**
Local	$\Delta = 0.2$	0.30
	$\Delta = 0.1$	1.17
	% change	290%
Neighborhood	$\Delta = 0.2$	1.54
	$\Delta = 0.1$	5.74
	% change	273%

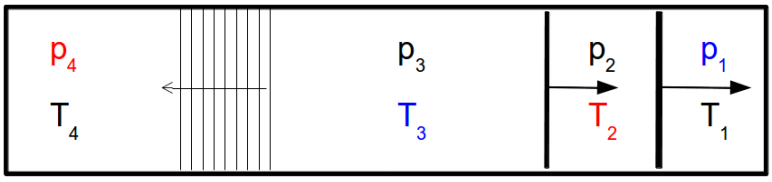
Table 13: Typical subiteration convergence and measured wall-clock time (in seconds) for interpolations in the inviscid convecting vortex case.

Test Case		**Average subiters./step**	**Average time/step**	**Total interp. time**	**% of Total walltime**
Local	$\Delta = 0.2$	8.61	0.0121	58.1	1.64%
	$\Delta = 0.1$	11.5	0.0332	159	0.69%
	% change	33.6%	174%	174%	-57.9%
Neighborhood	$\Delta = 0.2$	8.58	0.104	498	13.18%
	$\Delta = 0.1$	11.1	0.270	1,300	6.47%
	% change	29.3%	160%	160%	-50.9%

5.3 Shock Tube

A shock tube studied to demonstrate the application of thin-pate spline (TPS) and Wendland C^2 (W2) interpolation methodologies in a compressible flow with discontinuities.

(a) $t = 0$



(b) $t > 0$

Figure 52: Shock tube with a high and low pressure fluid (left and right, respectively) separated by a membrane; at time $t = 0^+$, the membrane is broken to form a right-traveling shock and left-traveling expansion wave (for $p_4 > p_1$). Relatively high, average, and low values are indicated by red, black, and blue text colors.

5.3.1 Theory

Shock tube physics may be described by inviscid, isentropic fluid mechanics of an ideal, calorically perfect gas. A diagram of the shock tube setup (Fig. 52) is adapted from Ref. [127]. A high and low pressure fluid, with values p_4 and p_1, are initially separated by a membrane or diaphragm; when ruptured, a right-moving shock wave is formed for $p_4 > p_1$. From a computational fluid mechanics perspective, the membrane is imaginary, and the experiment can be replicated by setting appropriate initial conditions throughout the computational domain. The problem is completely defined by the specified pressure ratio and temperatures.

There are several distinct physical features that can be resolved computationally

and also described by analytical solutions, providing an opportunity for validation. The first feature is the supersonic shock wave that is defined by a pressure discontinuity between regions 1 and 2. Also moving toward the right (albeit more slowly) behind the shock wave is a temperature discontinuity between regions 2 and 3, known as a contact surface. The cause for this feature is the pressure discontinuity from region 1 to 2 corresponding to an increase in temperature in region 2. Rapid acceleration and expansion of fluid causes a decrease in temperature on the other side of the surface in region 3. Finally, the pressure and temperature in region 3 and 4 are smoothly blended by a series of left-moving expansion waves.

The change in density is related to the change in wave speed by considering the conservation of mass in the moving wave frame of reference:

$$\frac{d\rho}{\rho} = -\frac{dV}{a}. \tag{76}$$

Then, for an isentropic, ideal gas (constant $\frac{p}{\rho^\gamma}$):

$$\frac{dp}{p} = \frac{\gamma}{\gamma - 1}\frac{dT}{T}. \tag{77}$$

By the definition of speed of sound ($a^2 = \left.\frac{\partial p}{\partial \rho}\right|_s$) is

$$\frac{da}{a} = \frac{1}{2}\frac{dT}{T}. \tag{78}$$

Assuming that density is only a function of pressure and temperature ($\rho(p,T) = \frac{\partial \rho}{\partial p}dp + \frac{\partial \rho}{\partial T}dT$),

$$\frac{d\rho}{\rho} = \frac{dp}{p} - \frac{dT}{T}. \tag{79}$$

This expression may be combined with Eqn. 76-78 to yield

$$V + \frac{2}{\gamma - 1}a = R_1 \tag{80a}$$

$$-V + \frac{2}{\gamma - 1}a = R_2 \tag{80b}$$

for left and right moving waves, respectively. The sign change in Eqn. 80b arises from a change of reference frame in Eqn. 76. The constants in Eqn. 80 are equivalent to the Riemann invariants presented in Eqn. 81 that describe the hybrid interface boundary condition (Section 6.1.2). The remainder of the problem may be described by considering the conservation of mass, momentum, and energy across the moving shock.

5.3.2 Interpolation Constraints

Due to the potential of overshoot in the vicinity of high gradient regions (for example, see Fig. 37 in Section 4.4), the use of higher-order mapping techniques can destabilize a solution if unchecked. For example, when interpolating the solution around a step change (discussed in Section 4.4), over- and under-shoots of over 10% were observed for the RBF methods (Table 6 in Chapter 4). These are manifestations of Gibb's phenomenon which have also been observed by other researchers [61]. In the shock tube, if the density field was initialized to have high and low density regions of $\rho = 10$ and 1, then a 10% undershoot results in a density of approximately 0.1, and the solution approaches that of a vacuum. For a larger density ratio, the problem can quickly become non-physical. Typically, realizability constraints are imposed within the flow solver during flux reconstruction to prevent non-physical solution values such as negative density, pressure, or total energy.

A clipping approach to prevent the creation of new extrema was found to be effective at maintaining numerical stability. Interpolated values outside of the range of local minima and maxima are replaced with the corresponding local minimum or maximum. This is the approach employed by another high-order overset interpolation method [65]. Consequently, this scheme still has the potential for increased accuracy while satisfying the maximum principle associated with linear mappings [24]. For instance, the step function studied in Section 4.4 would still achieve a 20% reduction

in gradient error but have no overshoot when applying clipping (Fig. 34). When clipping was not used in the shock tube problem, the solution immediately exhibited the Gibbs phenomenon with over- and under-shoot values on the order of one (for initial density and pressure ratios of 25). While this was only an error of 4% and less than the observed overshoot for the simpler two-dimensional step problem, the solution immediately became unstable (after a single solver iteration) due to locations in the flow field where the density was very close to zero or negative. Allowing a smaller percentage of over- or under-shoot (less than 4%) provided marginal numerical stability but did not improve solution accuracy. Moreover, such specifications are ad hoc in nature and were avoided. Therefore, to eliminate uncertainties from the interpolated solution, local extrema were enforced for all simulations.

5.3.3 Experiment Description

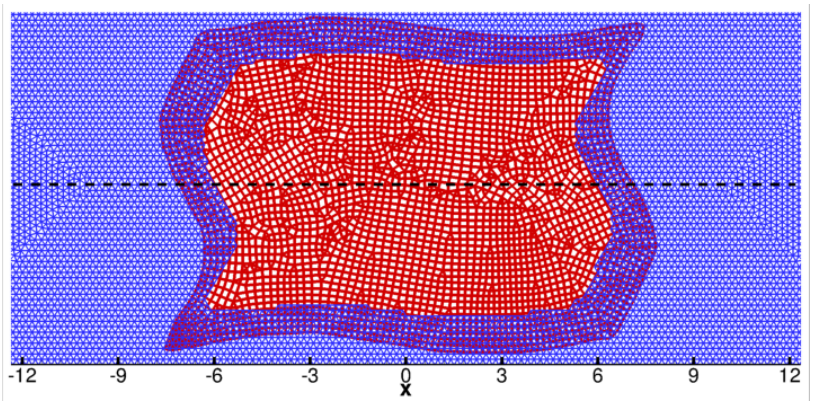


Figure 53: Unstructured overset meshes applied in the shock tube study. The dashed line denotes $z = 0$ where data were sampled.

To verify the robustness of the advanced mapping technique for general CFD applications, an unstructured version of the background grid was considered resulting in 40,944 nodes with 1,894 fringe locations. The mesh background mesh consisted of only tetrahedra, which is typical of input configurations for FUN3D. Only a baseline grid configuration with a spacing of $\Delta s = 0.2$ was studied (Fig. 53) to emphasize the differences in solution due to interpolation error. Grid refinement indicated that

the solution was more sensitive to spatial discretization error than interpolation error (discussed in Section 5.2.3). The time step size was set to 0.01 so that the fastest flow in the solution traveled less than a single cell length at each time step. To prevent oscillatory solutions in the region near shocks, the van Leer limiter formulation [128] was determined to be most successful in mitigating overshoots and limiting oscillations in the shock front. A single grid solution with the same grid spacing was referenced as the best possible overset solution.

The problem is fully defined for a fluid initially at rest by specifying the pressure and temperature ratios across the initial discontinuity, and temperature that defines the speed of sound for the medium. The inputs correspond to quantities depicted in Fig. 52, with a right moving shock. Reference values for the computations were chosen to be the right-hand side low value, and the left-hand side initial temperature and pressure were therefore known from the p_4/p_1 and T_4/T_1 relations [127]. Remaining properties (density and speed of sound) were computed from the pressure and temperature ratios: $\rho_4/\rho_1 = (p_4/p_1)/(T_4/T_1)$ and $a_4/a_1 = \sqrt{T_4/T_1}$.

Table 14: Numerical description of the shock tube problem.

Input Parameters	Value
Input pressure ratio, p_4/p_1	25.0
Input temperature ratio, T_4/T_1	1.0
Input reference temperature, T_1	303.15 K
Calculated shock pressure ratio, p_2/p_1	4.0471
Calculated shock temperature ratio, T_2/T_1	1.6083
Calculated shock speed, $M_s = V_1/a_1$	1.9005
Calculated contact discontinuity speed, V_2/a_1	1.1452
Calculated expansion wave speed (head), a_4/a_1	1.0
Calculated expansion wave speed (tail), V_3/a_1	-0.3743

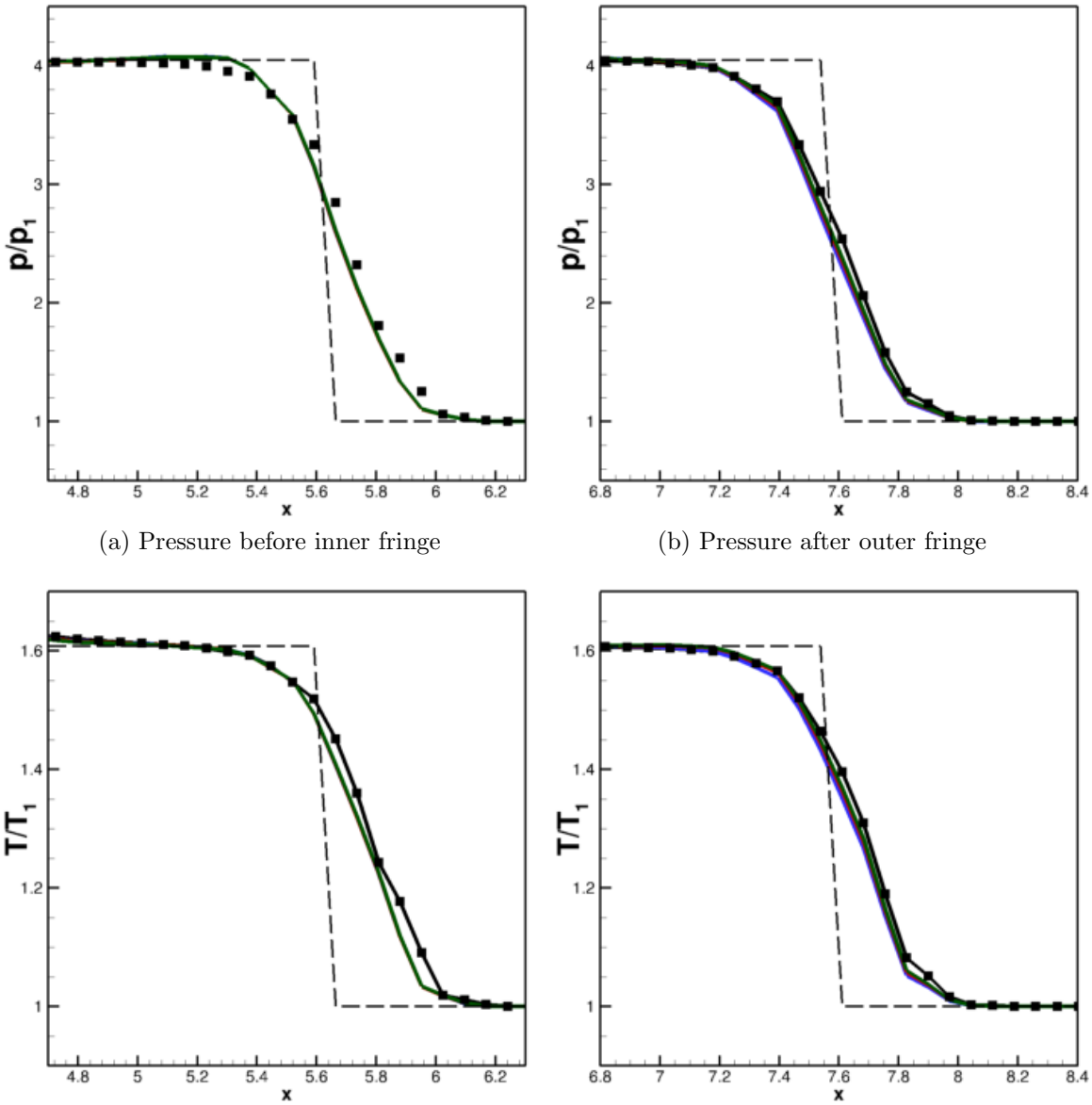
Solution parameters that set the initial solution field and drive the solution are listed in Table 14. The solution domain was partitioned (between initial high and low

density regions) at a slight offset from the center of the shock tube at $x = -\Delta s/2$. The offset allows nodes at $\pm\epsilon$ (with ϵ defined as machine precision) to be on the same side of the interface and results in a more planar interface between high and low pressure regions. For a right-moving shock, the most notable difference in solutions are expected to appear when the moving shock is between approximately $x = 6$ and $x = 7$ which corresponds to the $z = 0$ location of the inner fringe (on the background grid) and the outer fringe (on the patch grid). This occurs between simulation times $t = 3.2$ and 3.7. The moving shock traverses this overlap region in about 0.53 seconds (during approximately 50 solver iterations). In all cases, the clipping approach described in Section 5.3.2 was applied for numerical stability.

For the present study, the original interpolation stencil was expanded once outward to form an interpolation cloud; starting with the cell containing the target point, all nodes connected by an edge to each cell node are included. For this set of unstructured meshes, the number of donor points for each receptor is on average the same as the structured overset meshes with 24 donors for each donor-receptor transaction pair. The minimum and maximum number of donors for a target point is 18 and 28, depending on the number of nodes neighboring the enclosing cell.

5.3.4 Results

Prior to crossing the inner fringe into the overlap region, the shock solutions (in terms of pressure and temperature) are indistinguishable for the different interpolation techniques (Figs. 54a and 54c). It is not until after the shock front has completely exited the overlap region that differences between solutions emerge (Figs. 54b and 54d). The corresponding quantitative errors between the overset and single grid solutions are tabulated in Table 15. In all cases the differences between the overset RBF results and the single grid solution are less than or equal to the result with trilinear interpolation. When the shock is at the inner fringe, the TPS and W2



(a) Pressure before inner fringe

(b) Pressure after outer fringe

(c) Temperature before inner fringe

(d) Temperature after outer fringe

Figure 54: Evolution of shock wave front through the overset region, corresponding to values tabulated in Table 15. The overset solutions are represented by curves, the single grid solution as points, and the theoretical solution by the dashed line.

solutions are marginally closer by 0.1% and 0.3% in pressure, respectively, than the linear approach to the single grid solution. The temperature solutions are identical to within two decimal places (Table 15). At the outer fringe, the solutions are more differentiated, with the TPS approach showing 1.7% and 0.7% improvements over the linear estimates of pressure and temperature. The W2 approach is again closest to

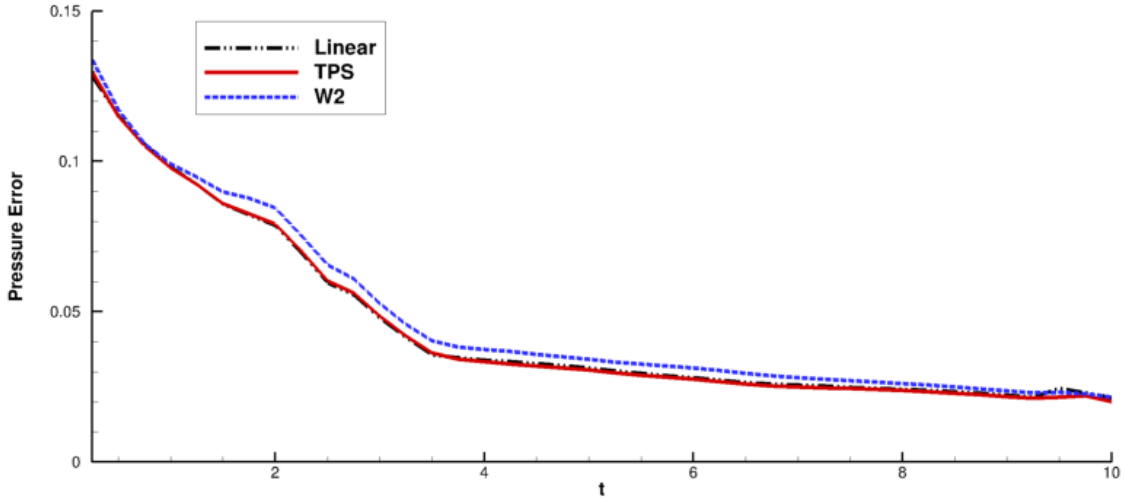
Table 15: Comparison of interpolated values in the shock tube at the inner and outer fringe centerline locations.

Solution Location	Methodology	p/p_1	T/T_1	p diff.(%)	T diff.(%)
Inner Fringe	Single Grid	3.335	1.519	-	-
	Linear	3.148	1.494	5.6	1.6
	TPS	3.150	1.494	5.5	1.6
	W2	3.158	1.495	5.3	1.6
Outer Fringe	Single Grid	2.943	1.464	-	-
	Linear	2.742	1.433	6.8	2.1
	TPS	2.793	1.444	5.1	1.4
	W2	2.819	1.449	4.2	1.0

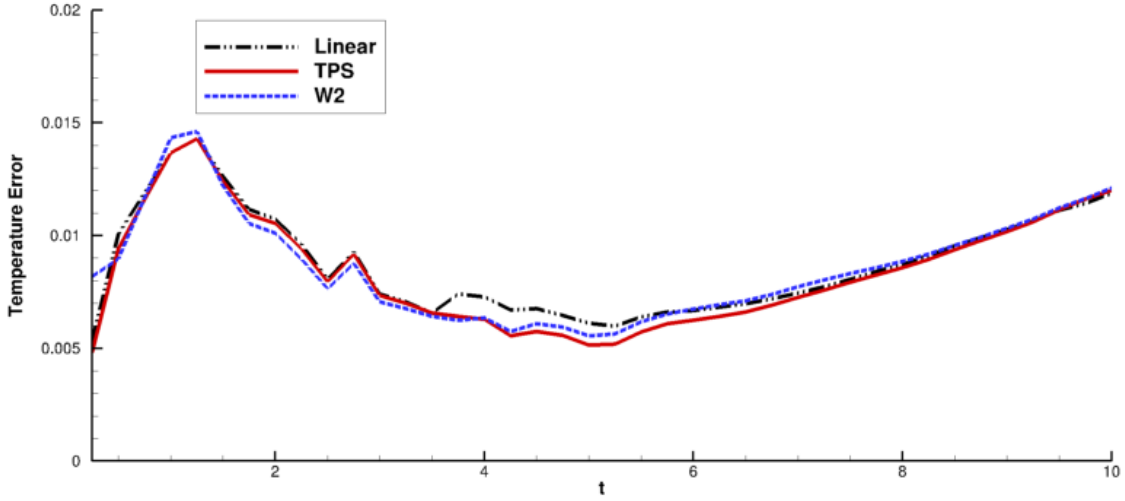
the single grid solution, with 2.6% and 1.1% less error than the linear approach.

The root-mean-squared errors in pressure and temperature offer another description of the overset errors in the solution throughout the computational domain over time (Fig. 55). Despite being more accurate near the shock front, the W2 approach accumulates more pressure error in the simulation after $t \approx 1.0$, prior to the shock reaching the overset boundaries (Fig. 55a). This error occurs from small over- and under-shoots throughout the shock tube away from the discontinuity, and persists throughout the simulation in the pressure solution. For most of the simulation time, the W2 pressure solution differed from the TPS calculated solution by 0.4–0.5% of p_1 (i.e., at most $0.005p_1$). Compared to W2, the TPS approach maintains accuracy consistent with the linear approach in terms of pressure.

When considering temperature, the solutions exhibit greater variation between the linear and RBF approaches (Fig. 55b). Since temperature is a higher-order solution quantity calculated from the quotient of pressure and density, it may be more sensitive to solution errors. A sudden increase in temperature error occurs at time $t \approx 2.5$, which is where the leading edge of the shock first reaches the inner fringe location at $x = 6$. For a theoretical shock speed (normalized by speed of sound) of $M =$



(a) Pressure Error



(b) Temperature Error

Figure 55: Evolution over time in RMS error of the overset solutions compared with the single grid solution for the shock tube simulation.

1.9, a planar shock would have reached the boundary at $t \approx 3.2$, indicating the effect of a smeared shock front as illustrated by Fig. 54. After the entire shock has passed through the overset fringe region, the effectiveness of the RBF approaches become more apparent (Fig. 55b) when the shock solution must be simultaneously calculated by the background grid and patch grid. Near $t = 4$, a maximum reduction in temperature error of roughly 14% is achieved. The temperature error for all three approaches converge after the shock passes through the second set of fringes and the

shock solution is once again solved solely by the background grid.

5.4 Ship Airwake

The advanced overset methodology has been further evaluated on the turbulent three-dimensional airwake of a model frigate. This application is motivated by the need to better characterize the operating environment in the vicinity of naval ships. Air vehicles typically encounter complex vortical structures shed from the bow of the ship and other ship structures, such as hangars, decks, turrets, cables, and parked air vehicles. For example, a helicopter trying to land on a ship deck aft of a hangar may have to descend through a shear layer emanating from the roof of the hangar or interact with a region of recirculating flow in the hangar wake.

5.4.1 Background

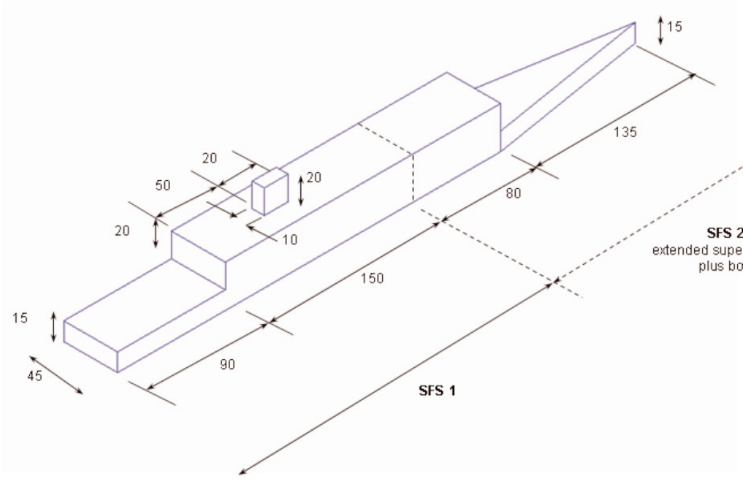


Figure 56: Geometries for the SFS and SFS2 configurations with dimensions in feet, from Ref. 129.

An international collaborative effort [130] has led to the development of a generic ship model known as the "simple frigate shape" (SFS) which has been extensively studied. Experimental data for this original model have been used for code validation in a number of previous studies [131, 132, 133]. A later iteration of this generic

model, known as the "simple frigate shape 2" (SFS2), introduced a more realistic, streamlined bow geometry (Fig. 56) and has since superseded the SFS. The SFS2 is a representative geometry of a ship topside developed by The Technical Co-operation Programme to allow standardized study of ship topside aerodynamics [129]. Test data from the 2 m $\times$ 3 m low-speed tunnel at the Aerodynamics Laboratory of the National Research Council in Canada [134] have been utilized by the research community [129, 132, 135]. New wind tunnel results of the SFS2 configuration have been obtained by the U.S. Navy from the Naval Surface Warfare Center Carderock Division (NSWCCD) and are available for additional validation. Recent efforts have studied a full-scale SFS2 in free air in addition to a 50th-scale model in the NSWCCD test section [136]. To determine the effects of the wind tunnel walls, simulations with an identical ship surface mesh have been considered under free-air conditions. 70.6 ft/s headwind conditions are simulated at a computational time step equal to the experimental sampling rate, 0.0005 s per step. The ship is modeled with viscous surfaces and the far field is modeled with a Riemann invariant condition. The sea boundary is considered to be part of the far field. This does not model the interface between air and water and facilitates convection of the ship wake downstream from the ship stern [136].

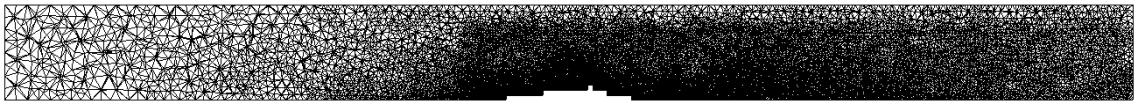
The overset approach permits simulation of these conditions using identical or similar grids as the previous simulations modeling a ship in the NSWCCD test section using a single grid. After creating a single background grid, a variety of operating conditions in future analyses (e.g., at different headings and/or with ship motion) may be simulated without the need to generate additional computational grids. Rather the near-body grid can be arbitrarily repositioned with respect to the background mesh. Assembly of the component grids (described in Section 5.4.2) was performed with Suggar++ to remove points from the background grid near and interior to the ship.

5.4.2 Grids

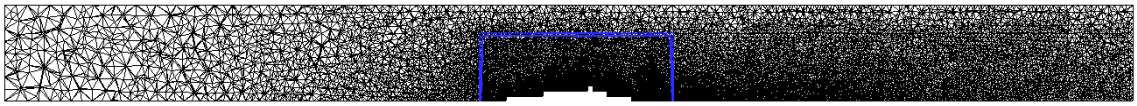
Table 16: Unstructured grids used in the ship airwake study

Grids	Nodes	Fringes
SFS2, single grid	2,668,155	-
SFS2, overset grids	3,020,415	54,958

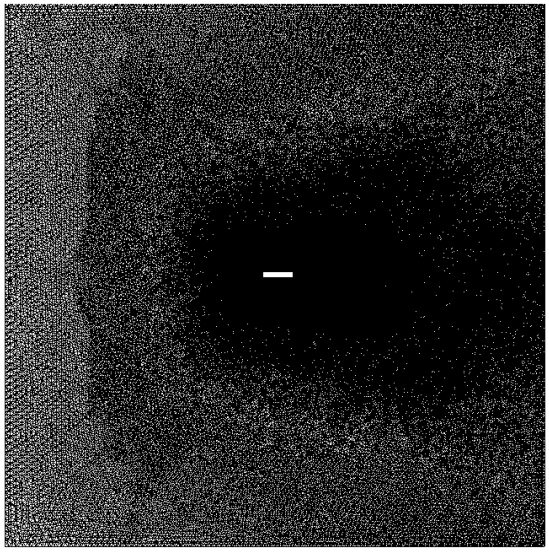
Following prior work [136], free-air grids were created from an existing grid which modeled the SFS2 in the NSWCCD wind tunnel test section. Point clustering on the viscous floor, ceiling, and side walls were removed for the free-air case. Since a URANS simulation is able to provide a large quantity of flow-field data not available from experiment, a single grid configuration was created for reference as a best possible solution to compare with the overset solutions. A free-air configuration was created by oversetting the modified near-ship grid onto a background box grid extending outward 4.5 ship lengths in the streamwise and lateral directions, and extending 0.75 ship lengths normal to the sea plane (Fig. 57). The ship surface point distribution was kept identical in the single and overset grid configurations to maintain grid quality ($y^+ < 1$) and permit comparisons between grids. The specified far-field mesh spacings were also kept the same. To further increase grid similarity, the outer boundary of the near-body overset grid (the points highlighted in blue in Fig. 57) was included in the single grid model as a point source; this source geometry was not retained in the final simulation grid. Grid details are provided in Table 16. The overlap minimization feature within the Suggar++ grid assembler was applied to expand the hole cut in the background grid, thus reducing the amount of overlap between grids. This procedure removed $\mathcal{O}(10^5)$ points from the background grid, shifting fringe point locations away from solid boundaries and enabling data transfer in regions where the mesh resolution is comparable between the overlapped grids [56].



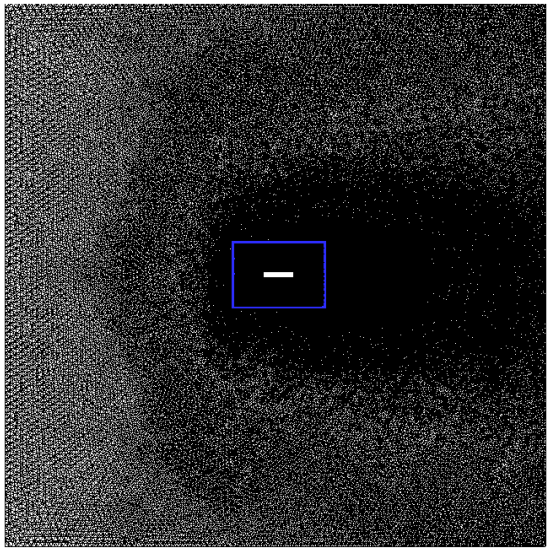
(a) SFS2 single grid, viewed from port side



(b) SFS2 overset grids, viewed from port side



(c) SFS2 single grid, viewed from above



(d) SFS2 overset grids, viewed from above

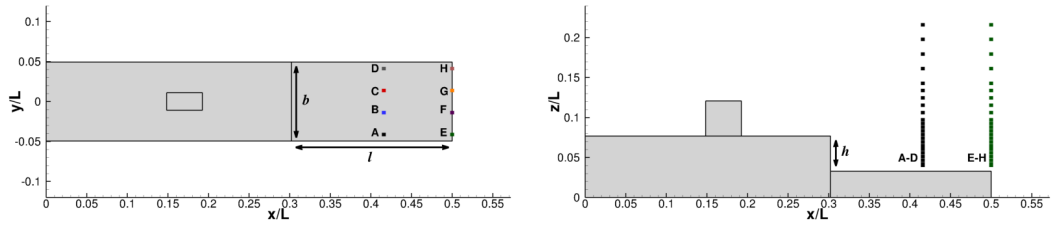
Figure 57: Single and overset grid configurations for the simple frigate shape 2 (SFS2) model. Side view is sliced through ship centerline, top view is sliced at hangar height. Near-ship grid fringe points are colored blue.

Table 17: Fast response probe (FRP) measurement locations on the 1/50th scale NSWCCD model. Streamwise positions are normalized by deck length (ℓ) while lateral positions are normalized by ship beam (b).

Location	x/ℓ	y/b
A	0.5741	-0.4167
B	0.5741	-0.1389
C	0.5741	0.1389
D	0.5741	0.4167
E	1.0000	-0.4167
F	1.0000	-0.1389
G	1.0000	0.1389
H	1.0000	0.4167

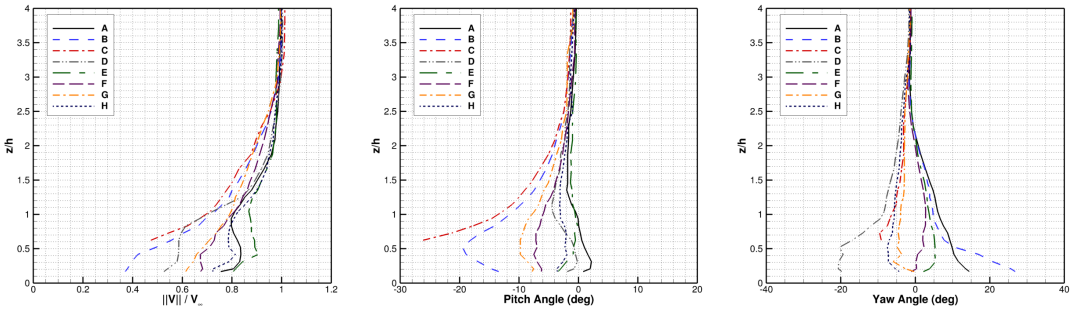
5.4.3 Comparison with NSWCCD Experiment

Table 17 identifies the eight locations at which fast response probes (FRPs) were placed in the experiment, illustrated in Fig. 58. Measurements were collected above the deck at heights ranging from 0.1667–4.1667h, where h is the height of the hangar. These measurements included velocity magnitude, flow pitch angle, and flow yaw angle whose averages are presented in the first, second, and third columns respectively of Fig. 59. These profiles are plotted as a function of height above the flight deck, normalized by the hangar height. The greatest differences between URANS calculations and experiment were observed below hangar height. The highly unsteady flow in this



(a) Top view (b) Side view

Figure 58: Measurement locations for the headwind NSWCCD case; distances are relative to ship center and normalized by ship length.



(a) Velocity magnitude, experiment (b) Pitch angle, experiment (c) Yaw angle, experiment

Figure 59: Averaged velocity magnitudes, pitch angles, and yaw angles measured in the NSWCCD experiment [136] (with h being the hangar height and $z/h = 0$ corresponding to the flight deck).

region is influenced by the separated, recirculating flow in the wake of the hangar (described in Section 5.4.6). As a result, the flow trajectories are asymmetric despite headwind flow that is symmetric across the ship mid-plane (normal to the streamwise and normal directions). This asymmetry is observed in the experimentally measured flow yaw angles (Fig. 59c), and is apparent even after averaging over 16,000 data samples. These measurements spanned a period of 8.0 seconds and corresponded to a flow-traveled distance of over 60 ship-lengths. Experimental data were unavailable at probe location C (Fig. 58a) below $z/h = 0.625$ because the sampled data were outside the calibrated range of the transducer [136]. Therefore only data at $z/h \geq 0.625$ have been utilized in the present analysis.

In all cases, experimental trends are reproduced for all measured variables (Fig. 60), most notably above the height of the hangar ($z/h = 1.0$). For example, the free-stream velocity magnitude ($||V||/V_\infty = 1.0$) tends to be recovered at a greater height above the deck at the inner probe locations (B, C, F, and G) compared with the outer probes (Fig. 60, first column). The downward flow pitch angle is greatest at the probes closest to the centerline (probes B, C, F, and G) and nearest to the hangar (probes B and C), as illustrated by the second column of Fig. 60. Finally, the flow yaw angle is largest at the outer front probes (A and D), indicated by the third column

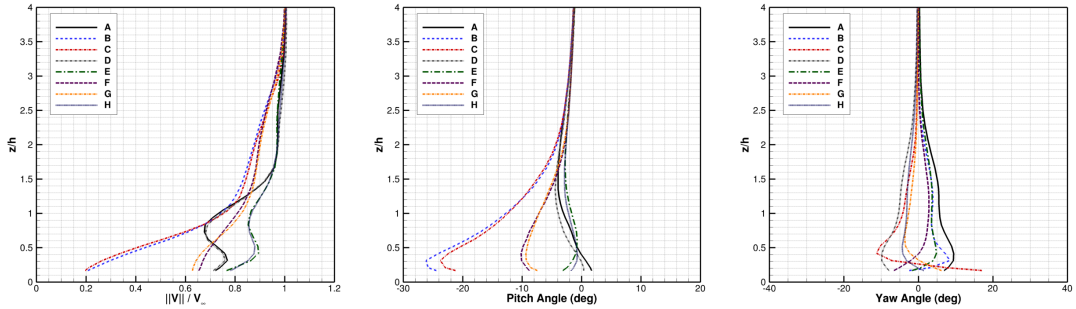
Table 18: Errors in the single and overset grid URANS calculations compared with the NSWCCD experimental data [136], averaged over all measurement locations.

Overset Method	$\|\|V\|\|/V_\infty$	Local Pitch Angle	Local Yaw Angle
Single Grid	0.0286	1.0618°	1.6863°
Overset, trilinear interpolation	0.0295	1.1348°	1.6646°
Overset, thin-plate spline	0.0276	1.1998°	1.8198°
Overset, Wendland C^2	0.0399	1.0452°	1.7531°

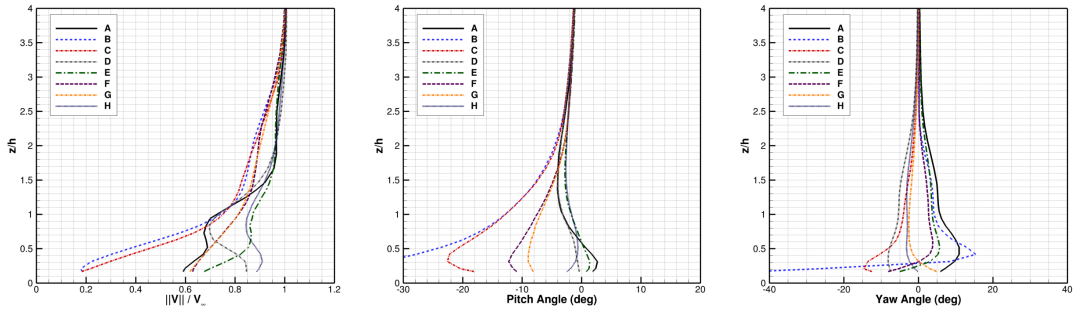
of Fig. 60.

Error profiles for the three sampled variables are presented in Fig. 61. Velocity magnitude errors are within 7% of the free-stream velocity above $z/h = 0.625$ (where data from all eight experimental probes is available) for all cases, including single and overset grids (Fig. 61, first column). Pitch and yaw angle errors are both within 4° (Fig. 61, second and third columns, respectively). Yaw angle errors above hangar height were small relative to the maximum observed yaw error ($\geq 30°$). The similarity in error profiles for the single and overset grid configurations suggest that the computational errors in this test case are not strongly influenced by the chosen overset methodology.

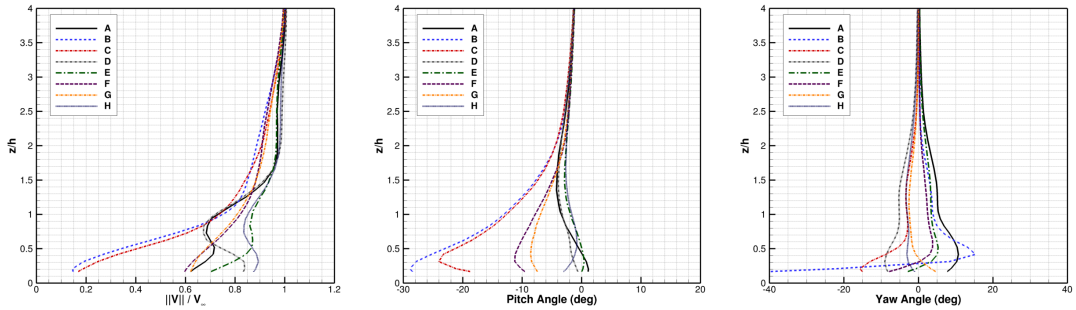
Errors averaged over all eight measurement locations are tabulated in Table 18. The velocity magnitude error out of all cases is at most 4% of the free stream velocity (from the Wendland C^2 calculation). The lowest error in velocity magnitude was observed in the thin-plate spline calculation at the expense of small increases in flow trajectory error (less than 0.14° in both pitch and yaw angle error compared to the single grid calculation). Local pitch angles differed from experiment by a maximum of 1.2° (from the thin-plate spline calculation) compared to a maximum observed downward angle of over 20° (Fig. 59b). Local yaw angles differed from experiment by a maximum of 1.8° (also from the thin-plate spline calculation) compared to experiment where the maximum observed yaw angles were ±20° (Fig. 59c).



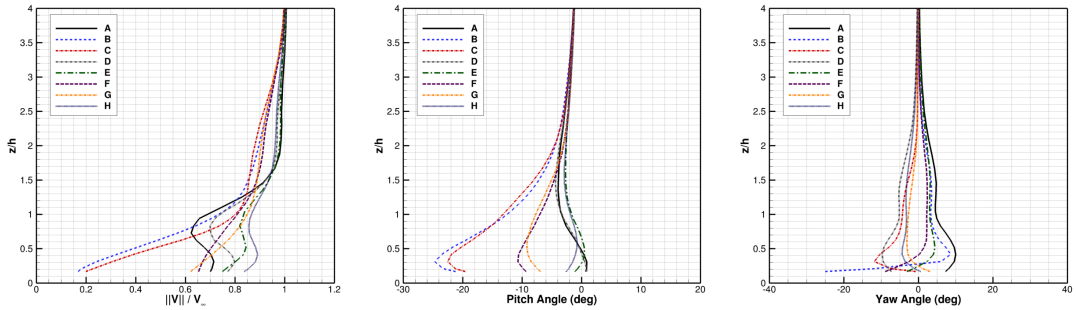
(a) Velocity mag., single grid (b) Pitch angle, single grid (c) Yaw angle, single grid



(d) Velocity mag., linear overset (e) Pitch angle, linear overset (f) Yaw angle, linear overset

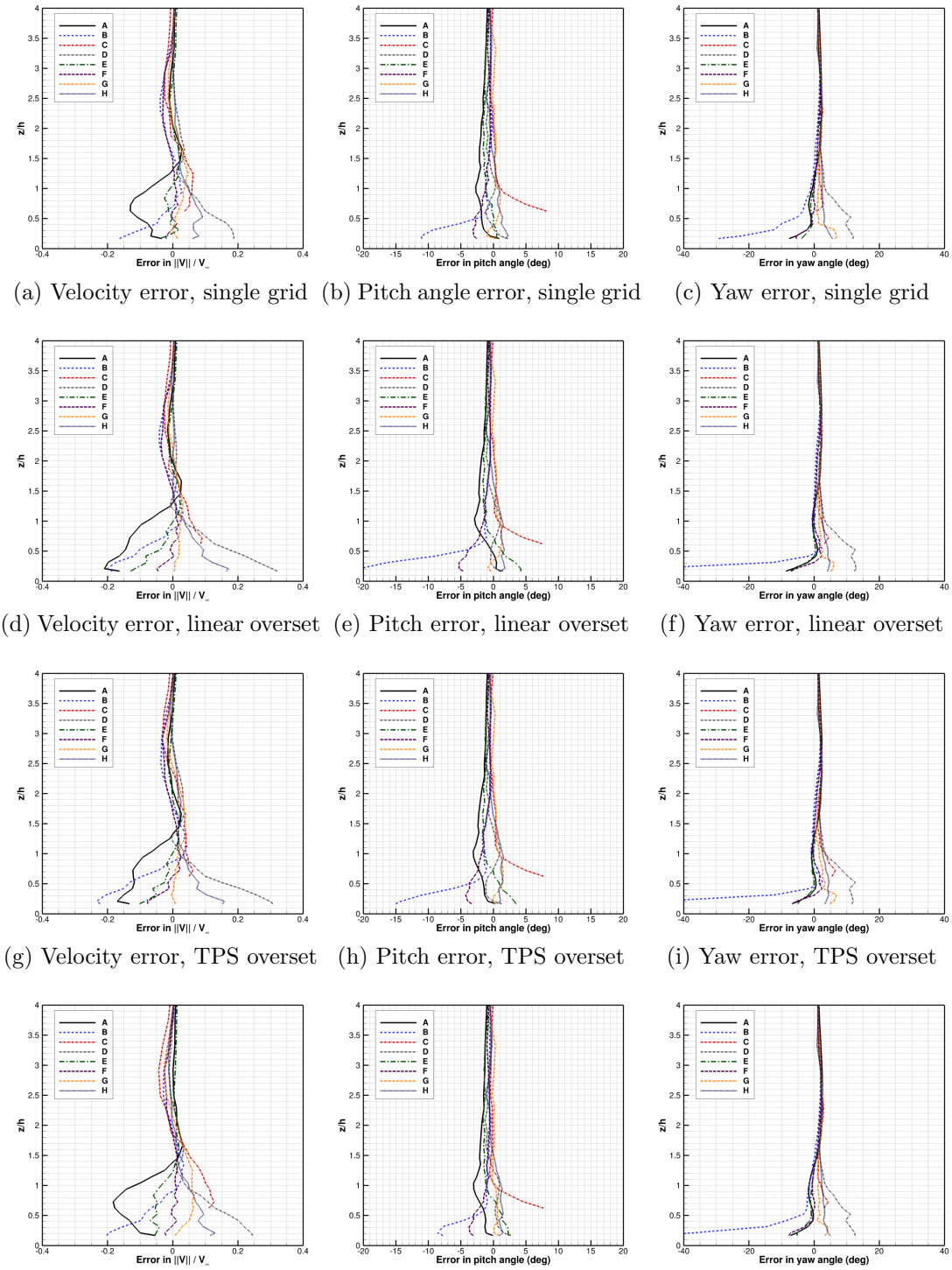


(g) Velocity mag., TPS overset (h) Pitch angle, TPS overset (i) Yaw angle, TPS overset



(j) Velocity mag., W2 overset (k) Pitch angle, W2 overset (l) Yaw angle, W2 overset

Figure 60: Averaged velocity magnitudes, pitch angles, and yaw angles at measurement locations (with h being the hangar height and $z/h = 0$ corresponding to the flight deck) for the headwind NSWCCD case, calculated with single grid and overset URANS.



(a) Velocity error, single grid (b) Pitch angle error, single grid (c) Yaw error, single grid

(d) Velocity error, linear overset (e) Pitch error, linear overset (f) Yaw error, linear overset

(g) Velocity error, TPS overset (h) Pitch error, TPS overset (i) Yaw error, TPS overset

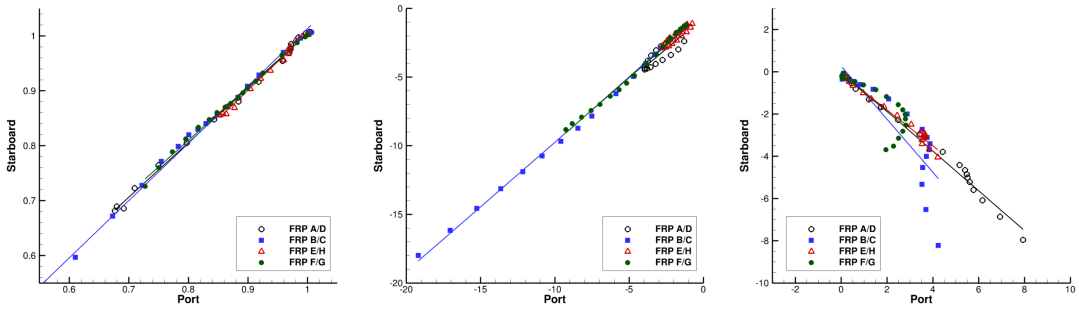
(j) Velocity error, W2 overset (k) Pitch error, W2 overset (l) Yaw error, W2 overset

Figure 61: Error in averaged velocity magnitudes, pitch angles, and yaw angles for the headwind NSWCCD case compared with experiment [136].

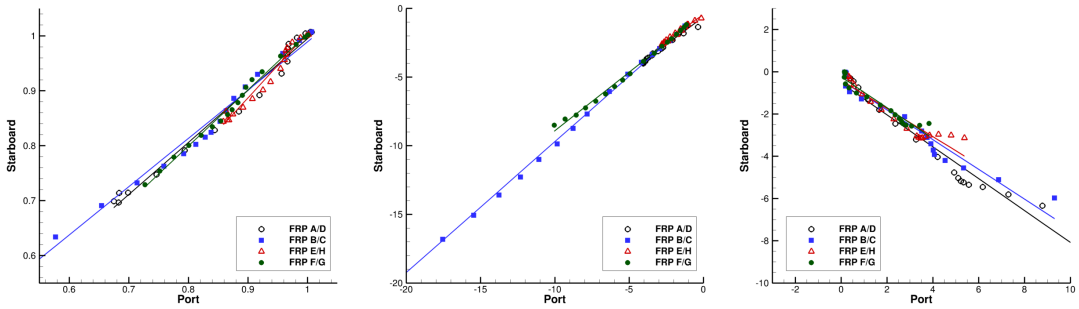
Table 19: Statistical relation (in terms of average slope and r^2 coefficient) between port and starboard-side measurements for the URANS simulations of the headwind NSWCCD case.

Overset Method	$\|\|V\|\|/V_\infty$		Local pitch angle		Local yaw angle	
	slope	r^2	slope	r^2	slope	r^2
Single Grid	1.01	1.00	0.92	0.91	-0.75	0.84
Overset, trilinear interpolation	1.01	0.98	0.85	0.99	-0.71	0.93
Overset, thin-plate spline	1.09	0.99	0.83	0.98	-0.65	0.78
Overset, Wendland C^2	0.75	0.98	1.01	0.97	-1.06	0.89

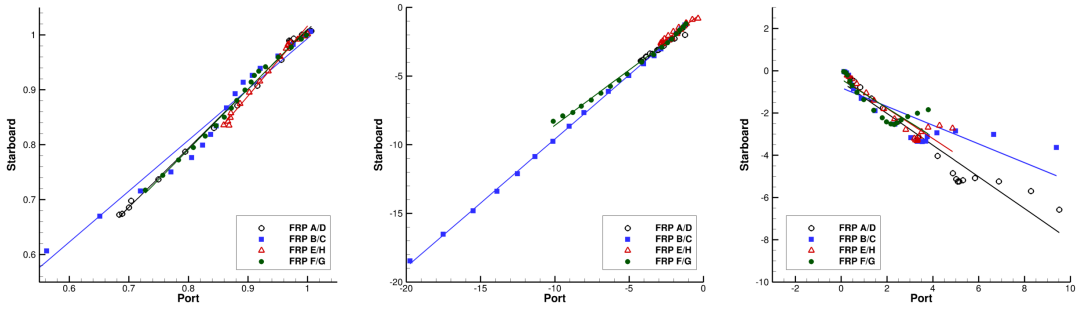
Pairing of the probed data from locations mirrored across the ship mid-plane, e.g., at locations A and D, B and C, and so forth (Fig. 58a) was observed. For a perfectly symmetric averaged flow solution, a cross-plot of velocity magnitudes at corresponding locations on the port and starboard sides of the ship would yield points distributed along a line with slope equal to 1.0. Similarly, if the pitch angles on both sides of the ship were perfectly correlated, the cross-plots of pitch angle would also yield a line with slope equal to 1.0. Conversely, the flow turns in opposite directions as it moves around the sides of the ship, therefore perfectly correlated yaw angles would give a cross-plot with slope of -1.0. Actual observations follow these trends (Fig. 62), with the least scatter (in terms of the Pearson r^2 correlational coefficient) observed in velocity magnitude and the most scatter in yaw angle for all methods (Table 19). Slopes were calculated by a linear least-squares fit of the data points. While the solution with Wendland C^2 interpolation has the velocity magnitude slope furthest from 1.0, the flow trajectory (dictated by pitch and yaw angles) exhibits the strongest symmetry. This result is similar to the averaged errors (Table 18) in which the Wendland C^2 calculations have slightly higher error in velocity magnitude but more accurately capture the local pitch angle behavior.



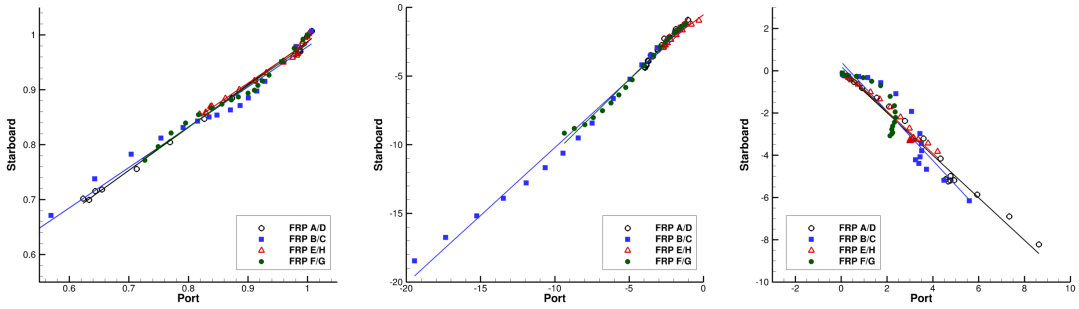
(a) Velocity magnitude, single grid (b) Pitch angle, single grid (c) Yaw angle, single grid



(d) Velocity magnitude, linear overset (e) Pitch angle, linear overset (f) Yaw angle, linear overset



(g) Velocity magnitude, TPS overset (h) Pitch angle, TPS overset (i) Yaw angle, TPS overset



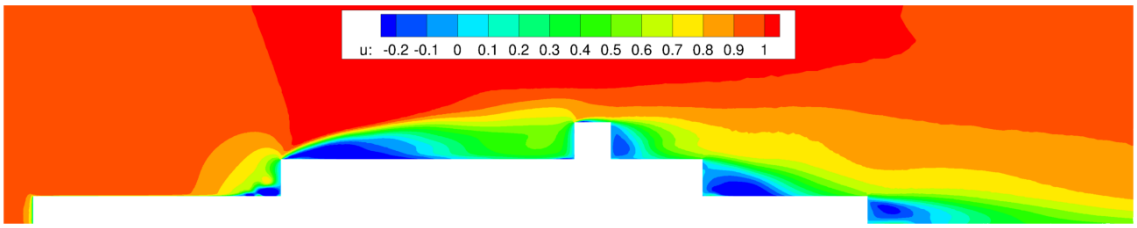
(j) Velocity magnitude, W2 overset (k) Pitch angle, W2 overset (l) Yaw angle, W2 overset

Figure 62: Cross-plots of averaged velocity magnitudes, pitch angles, and yaw angles between port and starboard measurement locations, for the headwind NSWCCD case.

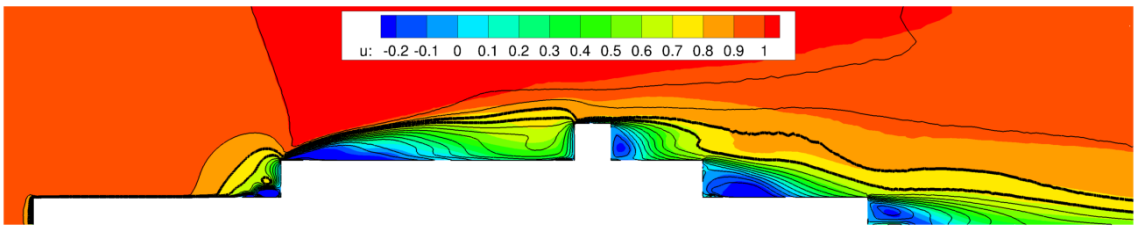
5.4.4 Averaged Flow Field

The sampled flow field was averaged over a period of 0.129 s, or the time required for the flow to traverse the distance of one ship length. Since there is no available experimental data with which to compare, the single grid solution is used as a reference solution. Figs. 63–66 depict the averaged flow fields for the linear and advanced overset approaches. Figs. 63 and 64 illustrate a sampling plane through the ship centerline viewed from the port side, while Figs. 65 and 66 illustrate a sampling plane at hangar height viewed from above the ship. Contours of streamwise velocity calculated by the single grid configuration have been overlaid to illustrate differences in the overset results.

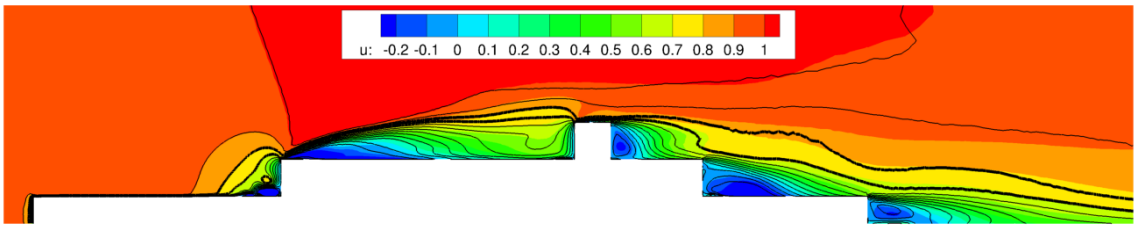
For all approaches (on single and overset grids), flow above the forward deck is identical. Regions of recirculating flow on the top side of the ship and behind the stack are of comparable extent for all methods. Below hangar height on the rear deck, the averaged flow behavior is also similar across all methods. For example, the extent of the region in which the flow is reversed is comparable; this is further discussed in Section 5.4.6. However, closer to the stern and near hangar height, differences in the flow are visible. Figure 63a–c illustrate the similarity in the velocity field near the stern between the single grid, trilinear overset, and thin-plate spline overset simulations above the stern with the velocity being between 0.7–0.8V_∞ (outlined by the bold contour lines). In contrast, overset interpolation with the Wendland C^2 function overpredicts the streamwise velocity at this location (Fig. 63d). This accounts for an increase in velocity magnitude error when applying Wendland C^2 interpolation equal to 1.1%V_∞ (Table 18). While there are errors in the location of the contour lines, the Wendland C^2 and thin-plate spline interpolation approaches both predict velocity fields qualitatively similar to the single grid calculation. In comparison, the overset solution with trilinear interpolation exhibits qualitative differences in the streamwise



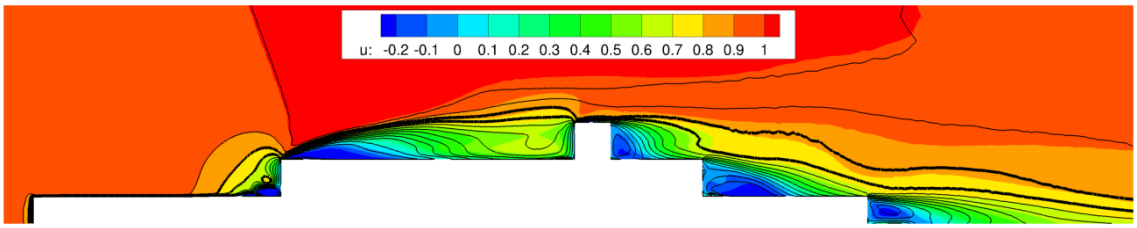
(a) Single Grid



(b) Trilinear Overset (color contours)



(c) Thin-Plate Splines Overset (color contours)



(d) Wendland C^2 Overset (color contours)

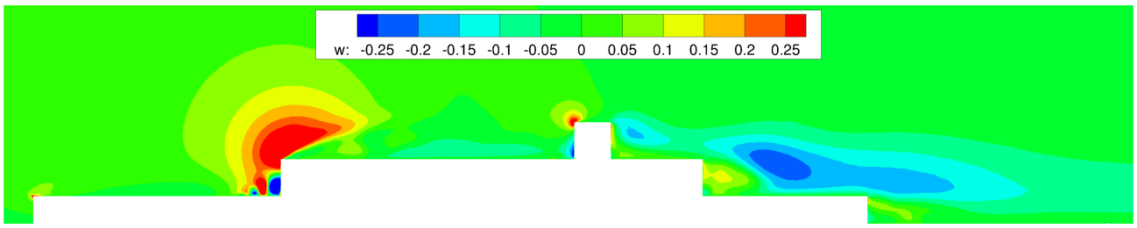
Figure 63: Averaged contours of streamwise velocity for single and overset grid configurations, viewed from the port side. In the overset cases, the black line contours represent the single grid solution.

velocity field above the flight deck above hangar height, which result in overpredictions of streamwise velocity by about $5\%V_\infty$.

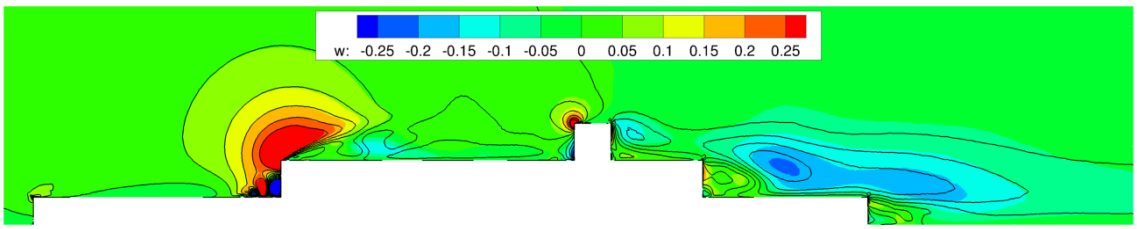
A side view of the normal velocity field is presented in Fig. 64. The velocity contours predicted by both single grid and overset methods are aligned upstream of the ship stack. Above the rear deck, the linear and thin-plate splines overset methods underpredict the magnitude of the normal velocity, with errors approximately equal to 3–$4\%V_\infty$. In comparison, the overset simulation with Wendland C^2 interpolation captured the normal velocity to within 1–2% of the free stream. This increase in accuracy accounts for the improved predictions of local pitch angle compared with experiment (Table 18).

When viewed from above, the streamwise velocity field above the front deck and on the top of the ship are nearly identical (Fig. 65). However, there are differences and asymmetries in the overset solution off the sides of the ship and in the wake of the hangar. Experimental observations discussed in the previous section suggest that the turbulent ship wake is not necessarily symmetric (across the ship centerline) even when averaging over long time periods. The largest error in streamwise velocity occurs in the hangar wake Wendland C^2 overset solution which differs from experiment by about $10\$V_\infty$, and also corresponds to the greater amount of yaw angle error (Table 18). To improve comparisons between methods, flow-field averages may be performed over additional data.

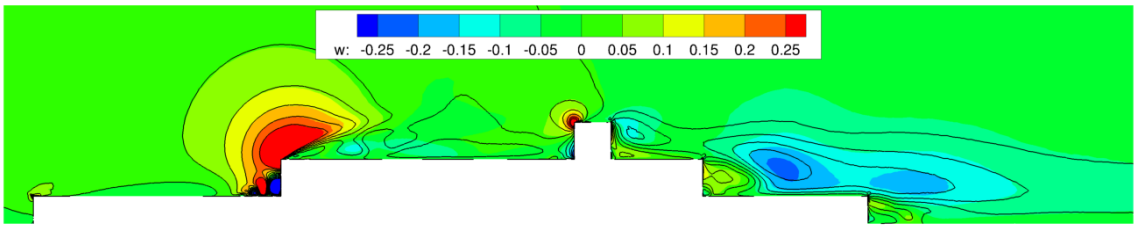
Lastly, lateral velocities have been considered from above (Fig. 66). There is agreement between all simulation methods in predicting how the flow turns outward around the body of the ship at the end of the forward deck. Where the flow reattaches on the side of the ship, the flow turns inward. The trilinear overset approach predicts the same location of the reattachment but underpredicts the magnitude of the lateral velocity. The Wendland C^2 result is closer to the single grid solution in lateral velocity magnitude, but incorrectly predicts the reattachment location. Finally, the thin-plate



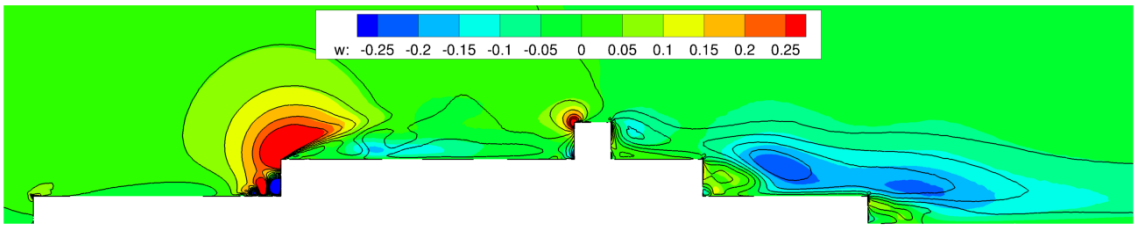
(a) Single Grid



(b) Triinear Overset (color contours)

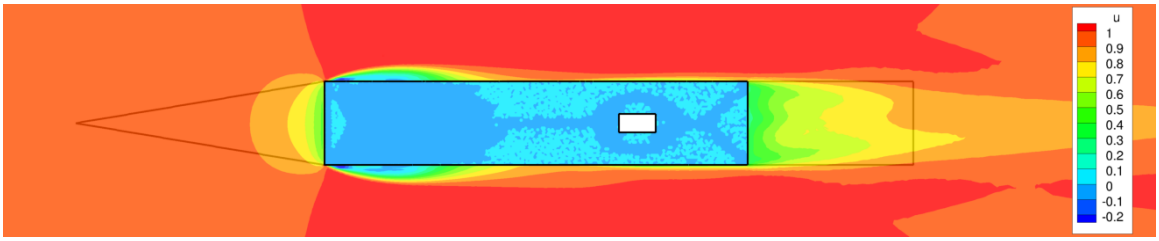


(c) Thin-Plate Splines Overset (color contours)

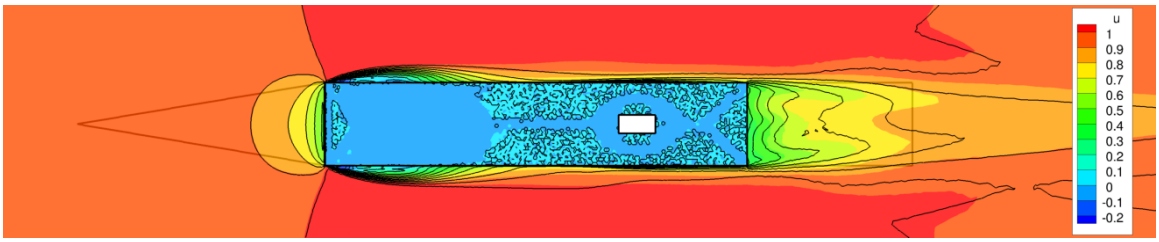


(d) Wendland C^2 Overset (color contours)

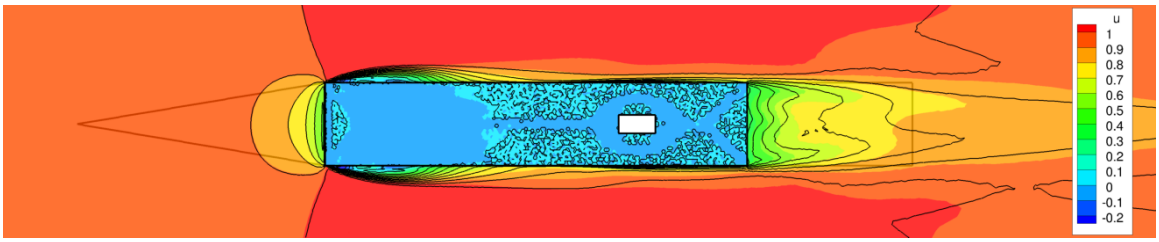
Figure 64: Averaged contours of normal velocity for single and overset grid configurations, viewed from the port side. In the overset cases, the black line contours represent the single grid solution.



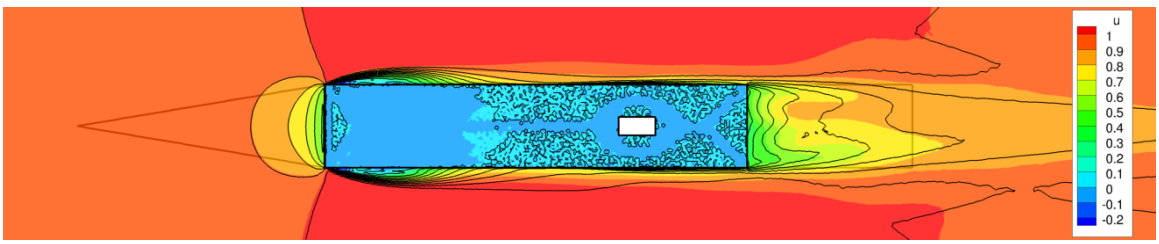
(a) Single Grid



(b) Trilinear Overset (color contours)



(c) Thin-Plate Splines Overset (color contours)



(d) Wendland C^2 Overset (color contours)

Figure 65: Averaged contours of streamwise velocity for single and overset grid configurations, viewed from above the ship. In the overset cases, the black line contours represent the single grid solution.

splines solution both predicts a lower lateral velocity magnitude and reattachment further upstream along the sides of the ship. These errors have a minimal effect on the flow aft of the hangar, where all overset simulations regardless of interpolation method overpredict the lateral velocity at the sides of the hangar by about $5\%V_\infty$.

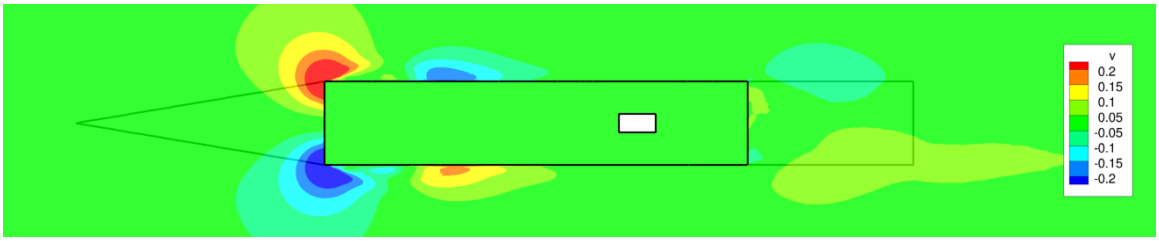
5.4.5 Wake Spectra

The wake flow field was sampled at hangar height, at streamwise locations corresponding to experimental measurement locations (Table 17). Velocity and turbulent kinetic energy data were collected at lateral positions spanning the ship width, extending half a ship width outward off both sides of the ship. The data were collected at the experimental sampling frequency of 2000 Hz, with 2048 data points included in the fast Fourier transform. Since unsteady experimental data were not available, the velocity and turbulent kinetic energy spectra for the overset simulations were compared with the single grid simulation (summarized in Table 20).

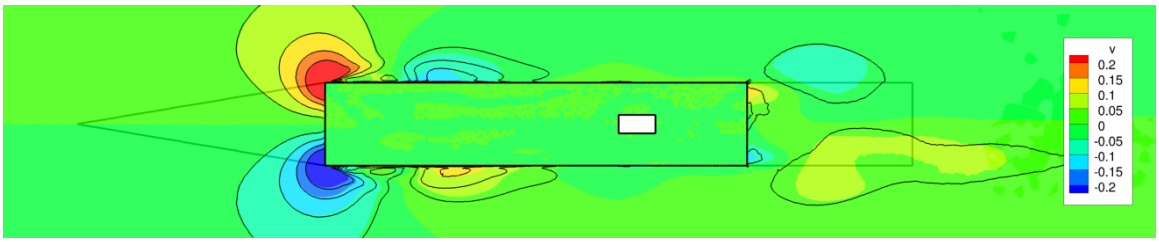
Table 20: Root-mean-squared (RMS) and maximum errors in velocity (V) and turbulent kinetic energy (k) spectra in the overset simulations compared to the single grid calculation.

Overset Method	V, RMS	V, max	k, RMS	k, max
Through measurement locations A–D				
Trilinear interpolation	1.11×10^{-4}	5.27×10^{-3}	3.16×10^{-9}	2.72×10^{-7}
Thin-plate spline	1.08×10^{-4}	4.34×10^{-3}	3.08×10^{-9}	2.69×10^{-7}
Wendland C^2	1.20×10^{-4}	5.22×10^{-3}	3.15×10^{-9}	2.17×10^{-7}
Through measurement locations E–H				
Trilinear interpolation	6.07×10^{-5}	2.30×10^{-3}	8.49×10^{-10}	6.32×10^{-8}
Thin-plate spline	5.92×10^{-5}	1.68×10^{-3}	7.76×10^{-10}	6.22×10^{-8}
Wendland C^2	6.54×10^{-5}	1.96×10^{-3}	8.20×10^{-10}	5.37×10^{-8}

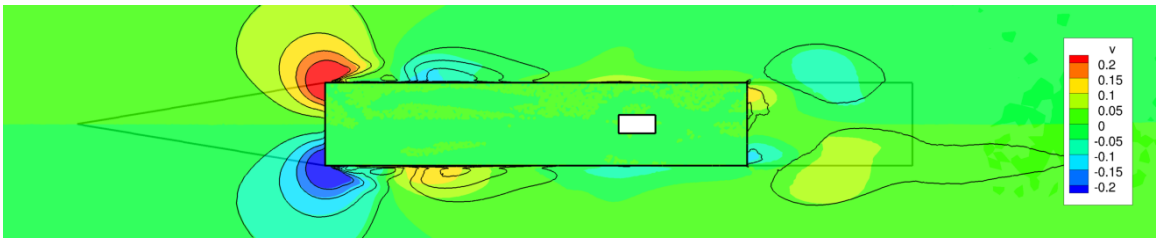
The resulting velocity spectra are plotted in Fig. 67 for the single grid simulation and overset grid simulations with linear and RBF interpolation applied. From the single grid solution (Fig. 67a and b), the most prominent frequencies appearing at



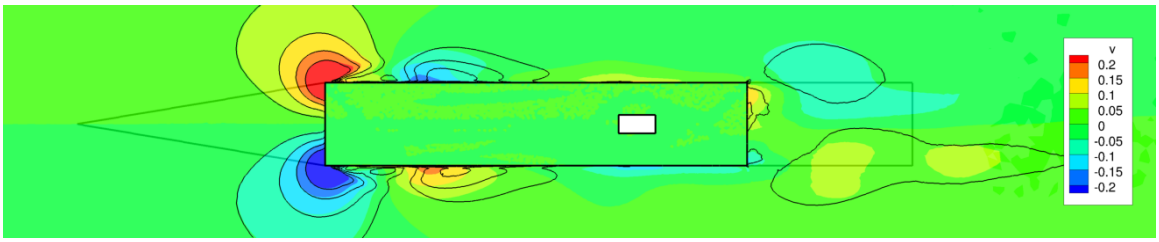
(a) Single Grid



(b) Trilinear Overset (color contours)



(c) Thin-Plate Splines Overset (color contours)



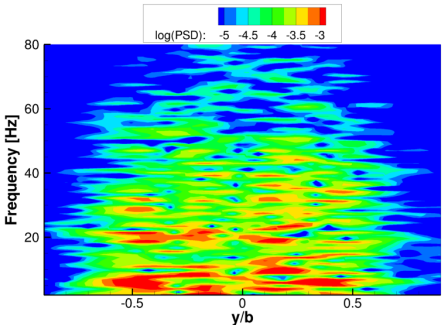
(d) Wendland C^2 Overset (color contours)

Figure 66: Averaged contours of lateral velocity for single and overset grid configurations, viewed from above the ship. In the overset cases, the black line contours represent the single grid solution.

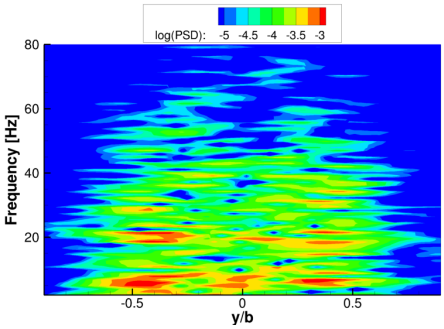
both sampled locations are near 5 and 20 Hz. These frequencies are present in all the overset simulations, but at different spectral densities. The sampling locations in the front row of fast response probes (FRPs), probes A–D, are immersed in the wake of the hangar and the ship stack. In contrast, the rear row of FRPs (probes E–H) is more fully immersed in the flow passing over the top of the ship (Fig. 63) that is canted downward and displaces the hangar wake. As a result, the influence of the hangar wake is less pronounced near the stern (at hangar height), illustrated by the reduced frequency content in the second column of Fig. 67.

Overset errors in the velocity spectra (compared to the single grid solution) are plotted in Fig. 68. The errors for all overset methods at both sampling locations are on the same order of magnitude as the actual calculated power spectral density (in comparison with Fig. 67). When applying thin-plate spline interpolation for overset interpolation, the maximum error observed in the velocity spectra is reduced by between 18–27% (Table 20). Wendland C^2 interpolation yields reductions in the maximum error of 1–15%. When considering the averaged error (defined by the root-mean-squared difference between the overset and single grid spectra), thin-plate splines interpolation also provides reductions in error of 2–3% while Wendland C^2 interpolation increases the average error by 8% at both sampling locations. This discrepancy is due to the variations in streamwise velocity distribution observed above the stern in Fig. 63d.

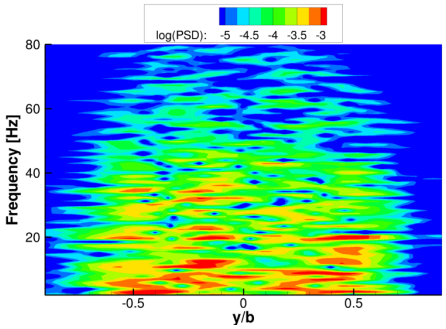
The turbulent kinetic energy (k) spectra (illustrated in Fig. 69) exhibit the same trends as the meanflow velocity spectra, namely that the frequency content is reduced at the downstream sampling location (through probes E–H) and that the overset errors have the same order of magnitude as the calculated power spectral density (Fig. 70). Moreover, the k spectra calculated from the overset simulations appear to have a lateral bias that is more pronounced than in the single grid simulation. Nevertheless, the RBF overset interpolation approaches were able to reduce both



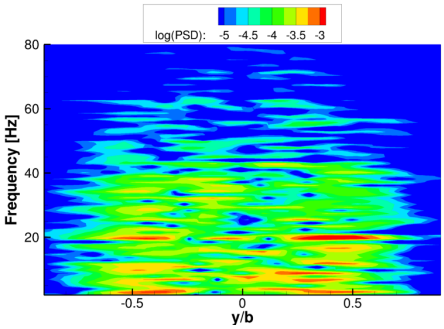
(a) Single grid, at measurement locations A–D



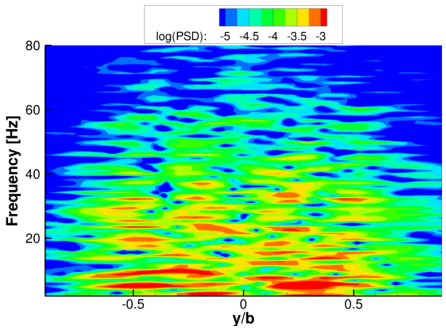
(b) Single grid, at measurement locations E–H



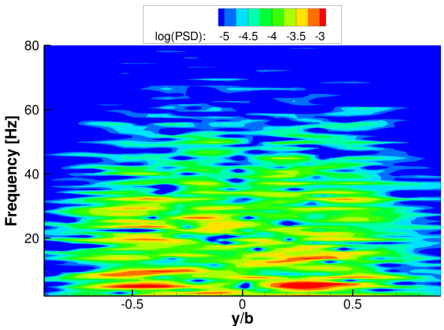
(c) Trilinear interpolation, at measurement locations A–D



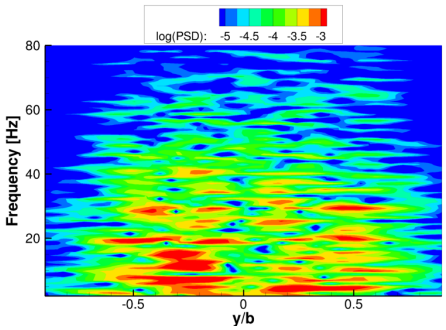
(d) Trilinear interpolation, at measurement locations E–H



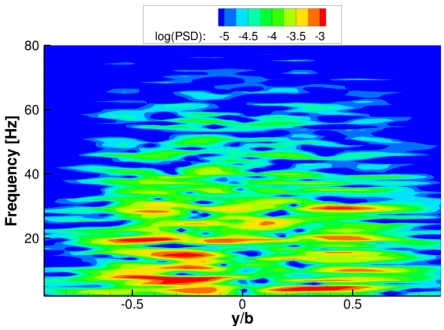
(e) Thin-plate spline interpolation, at measurement locations A–D



(f) Thin-plate spline interpolation, at measurement locations E–H

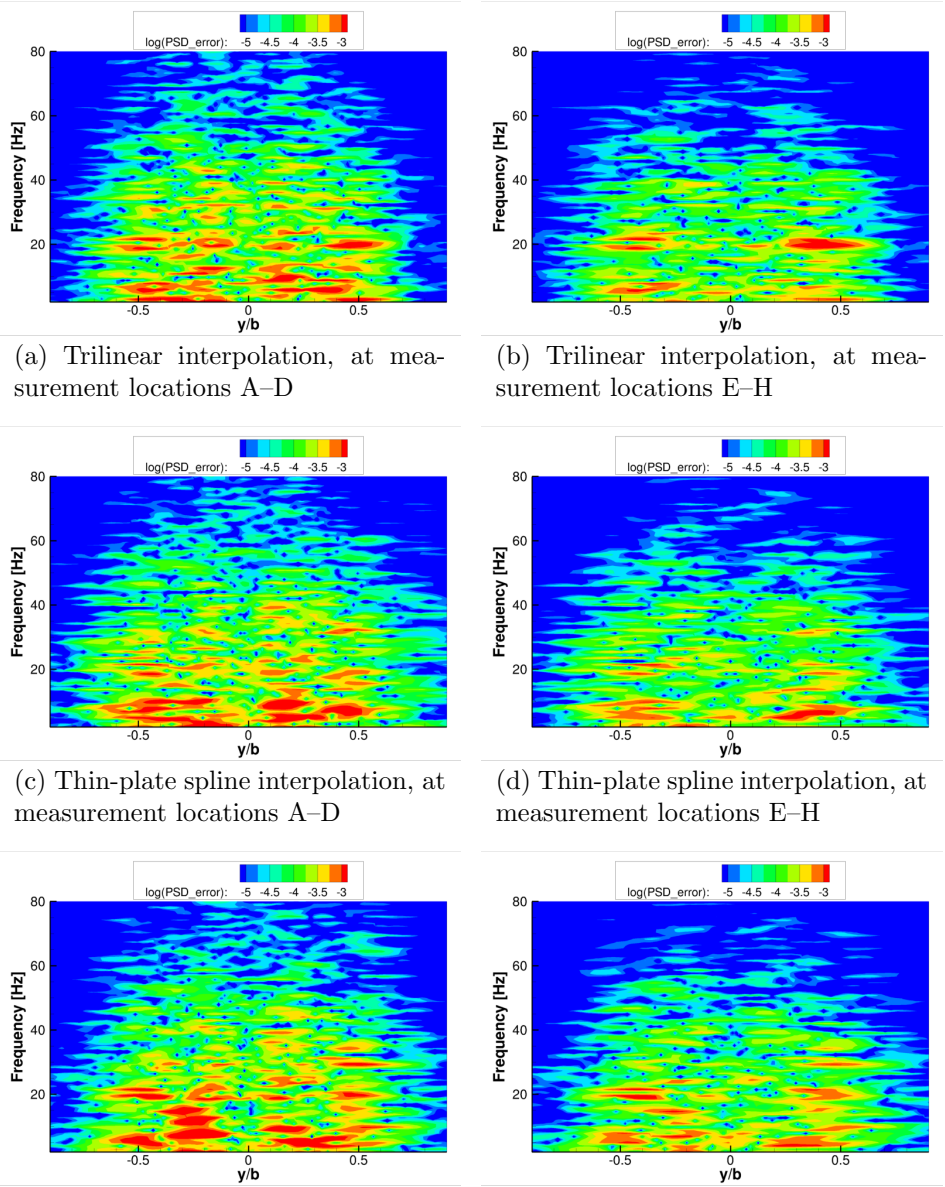


(g) Wendland C^2 interpolation, at measurement locations A–D



(h) Wendland C^2 interpolation, at measurement locations E–H

Figure 67: Wake velocity spectra calculated by linear and RBF overset interpolation methods. Data were sampled above the flight deck at experimental sampling locations corresponding to Fig. 58, with positions normalized by ship width.



(a) Trilinear interpolation, at measurement locations A–D

(b) Trilinear interpolation, at measurement locations E–H

(c) Thin-plate spline interpolation, at measurement locations A–D

(d) Thin-plate spline interpolation, at measurement locations E–H

(e) Wendland C^2 interpolation, at measurement locations A–D

(f) Wendland C^2 interpolation, at measurement locations E–H

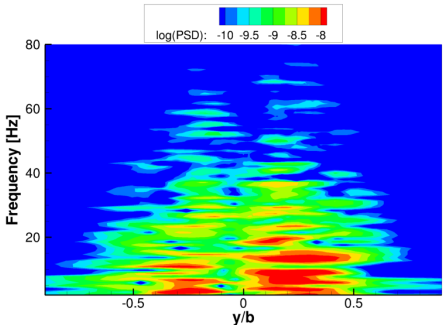
Figure 68: Errors in the wake velocity spectra calculated by linear and RBF overset interpolation methods in comparison with a single grid calculation. Data were sampled above the flight deck at experimental sampling locations corresponding to Fig. 58, with positions normalized by ship width.

average and maximum errors at both sampling locations (Table 20). Thin-plate splines overset interpolation reduced the maximum error by 1–2% and reduced the RMS error by 3–9%. Wendland C^2 interpolation reduced the maximum error by 15–20% and the RMS error by up to 3%.

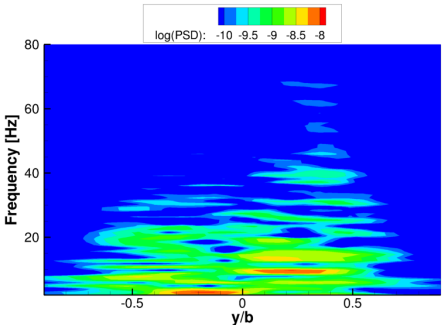
5.4.6 Attachment and Separation Locations

A flow recirculation region behind the hangar was observed in the original SFS experiment [133] and was suggested by the recent NSWCCD experimental results in Ref. 136. Within the present calculations, this recirculation zone corresponds to a region of separated flow that extends from the hangar up to approximately 60% of the flight deck length (ℓ), illustrated by the line contours at which the deck skin friction is zero (Fig. 71). To obtain a more precise estimate of the locations at which the flow separates and reattaches on the flight deck, the blue-colored contours from Fig. 71 were identified based on the criterion that the contour curve covered the largest portion of the flight deck in terms of the minimum and maximum x/ℓ. The positions of the point on this curve closest to the centerline, along with its two adjacent points on the curve, were averaged over the final second of simulation time. This included 2000 measurements during which the flow traversed a distance of 7.8 ship lengths. The positions at which the skin friction switches from positive to negative in the streamwise direction have been tabulated as the separation location (Table 21). In the same manner, the positions at which the skin friction switches from negative to positive along the center line have been tabulated as reattachment locations.

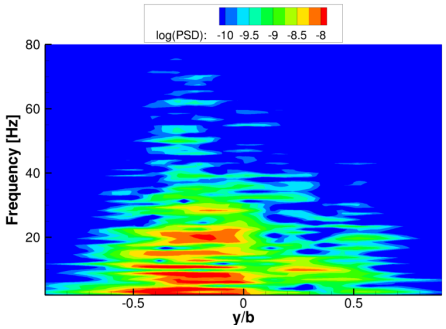
At the current flow conditions (70.6 ft/s, or 42 kts), the flow separates from the flight deck at approximately half a hangar height away from the hangar, and reattaches at a distance of approximately 2.5 hangar heights along the deck. The size of the recirculation region is approximately half the length of the deck for the SFS2 geometry. The separation location calculated from the overset simulation with



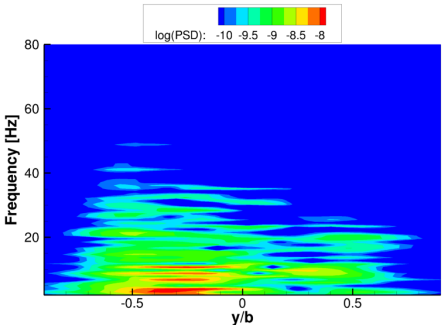
(a) Single grid, at measurement locations A–D



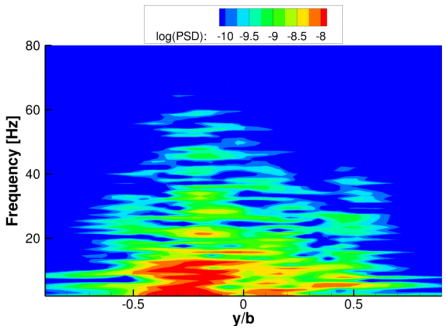
(b) Single grid, at measurement locations E–H



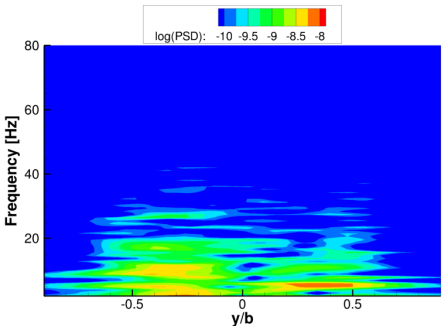
(c) Trilinear interpolation, at measurement locations A–D



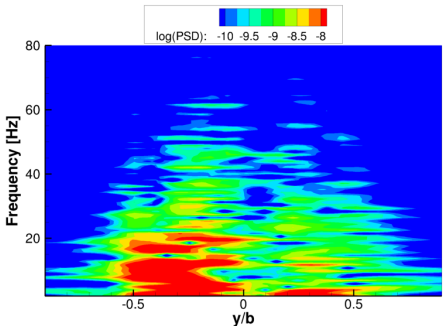
(d) Trilinear interpolation, at measurement locations E–H



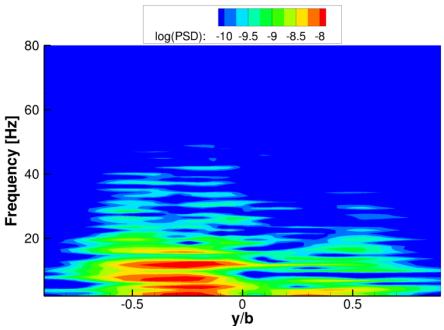
(e) Thin-plate spline interpolation, at measurement locations A–D



(f) Thin-plate spline interpolation, at measurement locations E–H

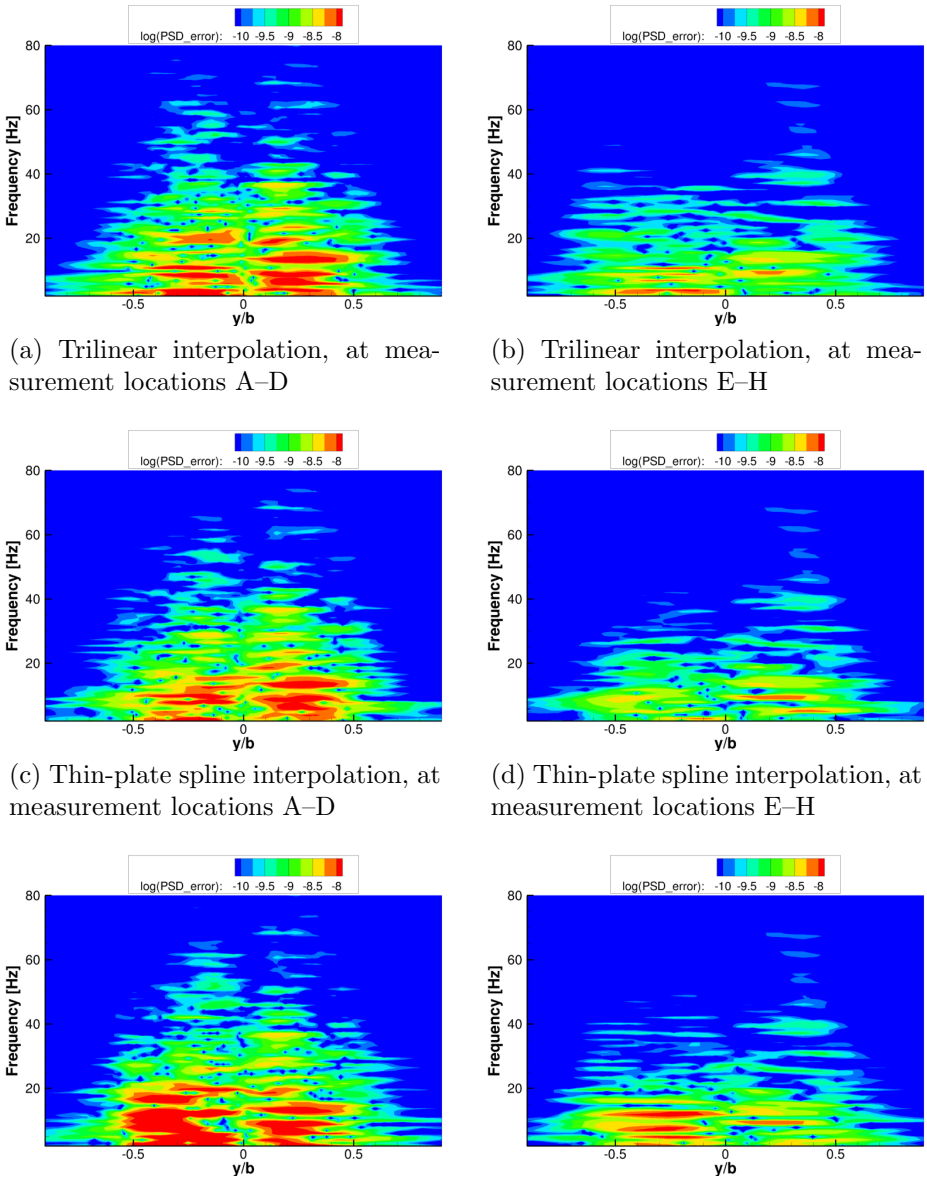


(g) Wendland C^2 interpolation, at measurement locations A–D



(h) Wendland C^2 interpolation, at measurement locations E–H

Figure 69: Wake turbulent kinetic energy spectra calculated by linear and RBF overset interpolation methods. Data were sampled above the flight deck at experimental sampling locations corresponding to Fig. 58, with positions normalized by ship width.



(a) Trilinear interpolation, at measurement locations A–D

(b) Trilinear interpolation, at measurement locations E–H

(c) Thin-plate spline interpolation, at measurement locations A–D

(d) Thin-plate spline interpolation, at measurement locations E–H

(e) Wendland C^2 interpolation, at measurement locations A–D

(f) Wendland C^2 interpolation, at measurement locations E–H

Figure 70: Errors in the wake turbulent kinetic energy spectra calculated by linear and RBF overset interpolation methods in comparison with a single grid calculation. Data were sampled above the flight deck at experimental sampling locations corresponding to Fig. 58, with positions normalized by ship width.

Table 21: Locations of separation and reattachment on the ship deck, normalized by hangar height h and deck length ℓ.

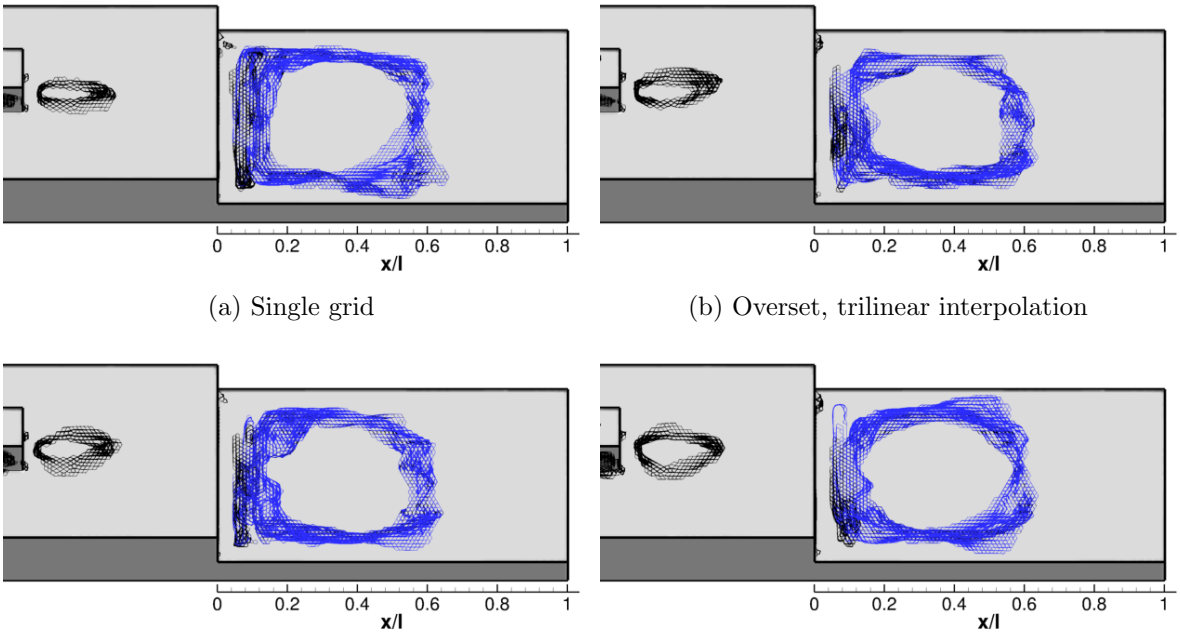
Overset Method	Separation Location		Reattachment Location	
	x/h	x/ℓ	x/h	x/ℓ
Single Grid	0.5247	0.1166	2.5646	0.5699
Trilinear Interpolation	0.4763	0.1059	2.6212	0.5825
Thin-plate spline	0.5175	0.1150	2.6099	0.5800
Wendland C^2	0.5171	0.1149	2.6271	0.5838

trilinear interpolation differs from the single grid solution by 9%. In comparison, both RBF overset interpolation methods predict a separation location within 2% of the single grid solution. In terms of reattachment location, estimates with linear and RBF overset methods all differ from the single grid solution by approximately 2%.

Simulations performed on the original SFS model noted errors in reattachment locations for the SFS model under headwind and yawed conditions at 40 kts [133], which may be compared to the present simulation at 42 kts. Wakefield et al. [133] observed that the recirculation region extends above the hangar roof and the flow reattaches at a stationary location, approximately $2.3h$ aft of the hangar. The URANS calculations predict a time-varying reattachment location near $2.6h$ that varies up to nearly a tenth of the deck length (Fig. 71), or $0.4h$. The averaged reattachment location from the single grid URANS calculation is closest to the SFS experiment, differing by 11%. The next closest calculation is with thin-plate spline interpolation (13% error) followed by the linear and Wendland C^2 calculations (both 14% error).

5.4.7 Cost

The ship airwake case was run in parallel on 256 processors. Table 22 reveals that the size of the interpolation problem increases eightfold for the cloud-based RBF approach when selecting a neighborhood of source points (e.g., all points that are connected to the cell enclosing the target point). Both single grid and overset calculations produced



(a) Single grid

(b) Overset, trilinear interpolation

(c) Overset, thin-plate splines

(d) Overset, Wendland C^2

Figure 71: Line contours at which skin friction is zero. Blue curves indicate the contours extracted from the hangar wake that were included in determining the average separation and reattachment locations on the ship deck. Dimensions are normalized by the length of the flight deck (ℓ).

solutions that were comparable to experimentally measured velocity magnitudes and flow trajectories, with expected correlations between port and starboard measurements. The thin-plate spline overset method required 6% more computational time (Table 23) but reduced errors in velocity magnitude by 6%. Applying Wendland C^2 interpolation also required approximately 6% more computational time, but reduced errors in the local pitch angle by 8%. In comparison with a single grid simulation, both RBF overset approaches reduced the maximum error in the wake velocity and turbulent kinetic energy spectra by up to 27% and 20%, respectively. Moreover, the error in predicted flow separation location on the deck was reduced by 7% when applying either RBF overset approach.

Table 22: Overset donor statistics for the SFS2 ship airwake study.

Overset Method	Avg.	Min.	Max.
Trilinear interpolation	4.001	4	6
RBF interpolation	32.13	17	47

Table 23: Cost in terms of averaged wallclock time per step for the test cases and methods presented.

Overset Method	Average cost/step [s]	% change
Trilinear interpolation	18.8474	-
Thin-plate spline	19.9706	6.0%
Wendland C^2	20.0179	6.2%

5.5 Treatment of Orphan Points

5.5.1 Problem Setup

Orphan points typically arise due to low-quality donor points. An example is the situation when one or more donors associated with a fringe point is itself also a fringe point (Fig. 72). The treatment of orphans is straightforward when applying a scattered data approximation technique. Since there is no requirement that the receptor point be located within a particular cell (as with standard mapping techniques) or that the donor points are connected (to form a stencil as required for polynomial-based techniques), points may be arbitrarily included in the data transfer to compensate for a lack of resolution in the immediate vicinity of the receptor. The RBF implementation was modified to allow dynamic resizing of the local interpolation neighborhood size to accommodate orphan points. In the original overset paradigm, the interpolation source points were chosen to be all the nodes neighboring the cell enclosing the target point. Smaller neighborhood sizes were shown to offer no improvement over linear interpolation (Section 3.3.1), while in the limit of increasing neighborhood size

(a globally supported interpolation) additional error reductions of 50–70% are possible between meshes of comparable point density (Fig. 15 in Section 3.3.1). However, to maintain feasibility in terms of computational cost, expanding the neighborhood of points beyond a single level of surrounding nodes has not been considered. For the current implementation, source points that are fringe points are removed from the interpolation cloud. If the number of non-fringe donors is less than or equal to the number of points comprising a single cell (e.g., eight for a hexahedral element), additional donors are sourced from neighbors of the neighboring nodes until the minimum donor requirement is met. Some sample clouds of source points are illustrated in Fig. 73. When a solution is needed for an orphan point (Fig. 73b), the selected cloud of source points does not encompass the target point therefore the data transfer is an extrapolation.

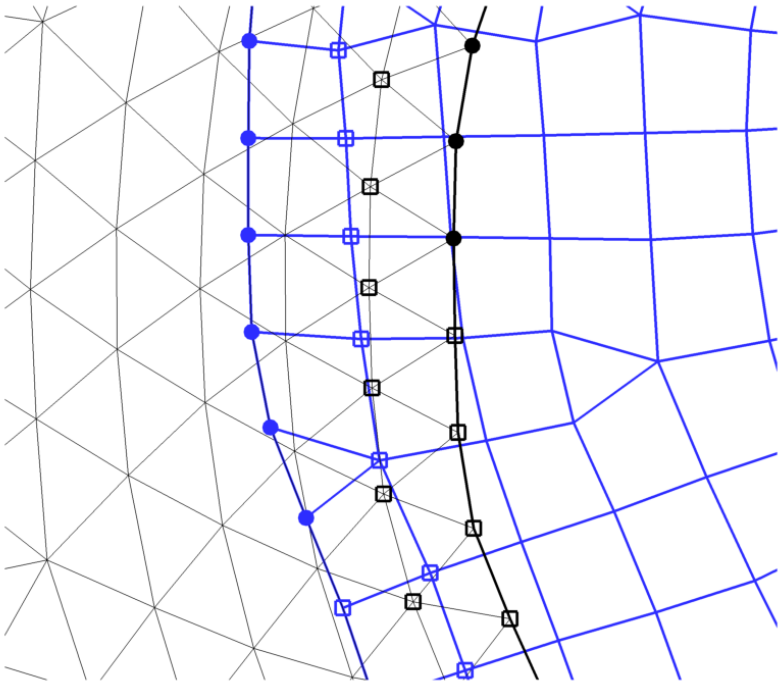
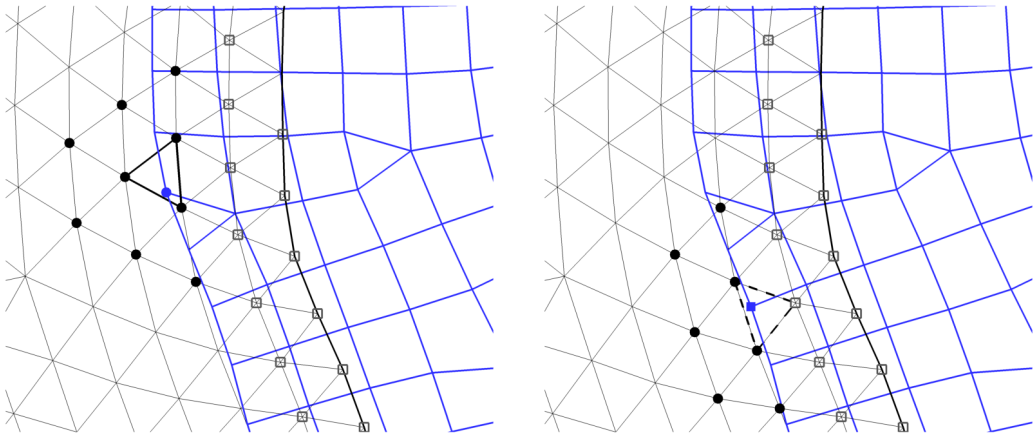


Figure 72: Close-up view of the overlap region between two overset meshes, with filled circles denoting normal fringe points and square symbols indicating orphans.

Overset simulations with trilinear interpolation performed data transfers with only the nodes of the cell enclosing the target point, which on average numbered between



(a) Normal fringe point (b) Orphan point

Figure 73: Clouds of source points in the vicinity of orphan points. Filled black circles denote donors while square symbols indicate orphans. The cell enclosing the receptor point is outlined on the black grid.

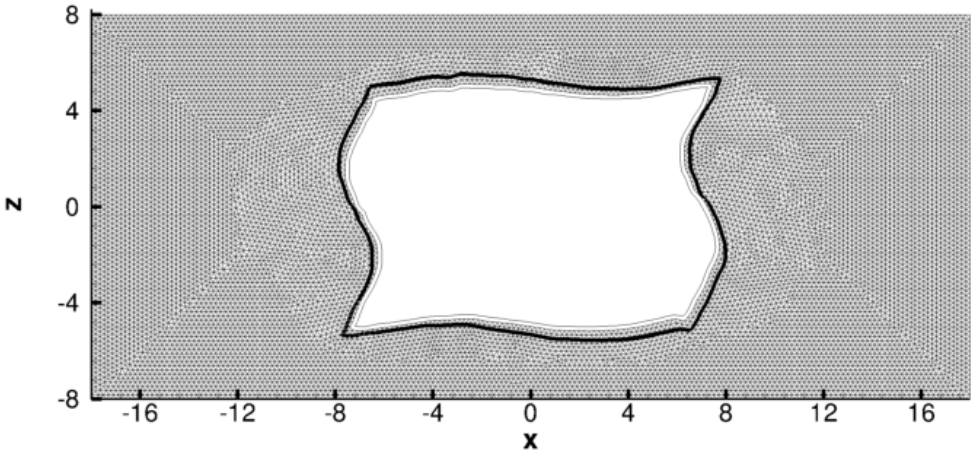
six and seven nodes (Table 24). For this two-dimensional configuration, computational cells are primarily triangles that have been extruded to form volumes for the three-dimensional flow solver. These triangles become six-noded pentahedra (in contrast to quadrilaterals becoming eight-noded hexahedra). The number of source points is fixed for this approach, regardless of whether orphan points are present. In comparison, the cloud-based RBF interpolations are based upon a neighborhood of 23–24 nodes on average, nearly a four-fold increase (Table 24). This indicates that most triangles are connected to nine other nodes on average (numbering 12 nodes on a two-dimensional plane). In the presence of orphan points, the number of available donors decreases by 19–38%. To increase accuracy over linear interpolation, the point selection algorithm requires that the number of source points not be less than the number of nodes within a single cell (in this case, six); the actual minimum number of points in a cloud was ten.

Table 24: Donor statistics for the vortex convection study.

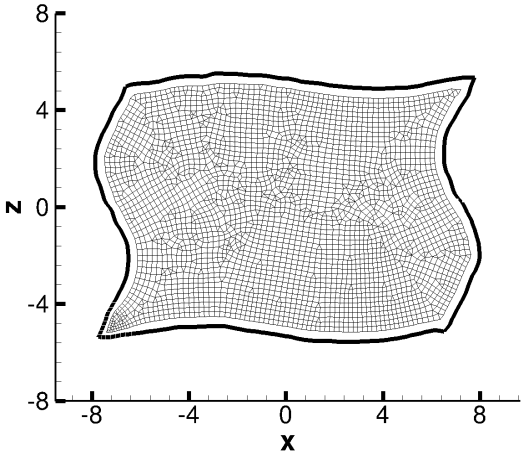
Overset Configuration	Mesh Spacing	Avg.	Min.	Max.
Trilinear interpolation	0.4	6.71	6	8
	0.2	6.84	6	8
	0.1	6.78	6	8
Trilinear interpolation	0.4	6.56	6	8
with orphans	0.2	6.84	6	8
	0.1	6.82	6	8
RBF interpolation	0.4	22.85	18	28
	0.2	24.02	18	28
	0.1	24.06	18	32
RBF interpolation	0.4	14.11	10	24
with orphans	0.2	16.23	10	26
	0.1	19.59	10	26

5.5.2 Grids

Additional unstructured test configurations containing orphan points were created based on the vortex test case, with the background meshes (Fig. 74a) modified to have an approximately 20% larger cutout region. Baseline, fine, and coarse meshes were considered, corresponding to isotropic cell sizes of $\Delta s = 0.2$, 0.1, and 0.4, respectively. In the baseline case ($\Delta s = 0.2$), the proximity of the overset boundaries reduces the extent of the overlap region and transforms over half of the fringe points (totaling 2,036) into orphans due to the poor quality of donors (numbering 1,152). This is illustrated in Fig. 75. Since these orphan points cannot be adequately resolved under the original overset interpolation paradigm, a region of uncertainty or an effective gap in the computational grid is formed. The number of total orphan points was kept approximately constant between the different mesh sizes to facilitate the analysis of overset errors due to orphans. On the coarse mesh ($\Delta s = 0.4$), the number of orphan points was an order of magnitude larger than the number of calculable fringe points. All grids considered in the present section are summarized in Table 25.



(a) Unstructured background grid with irregular cutout



(b) Irregular patch grid

Figure 74: Overset grid systems for advanced overset investigations with orphan points ($\Delta s = 0.2$); outlined region denotes the extent of the patch grid.

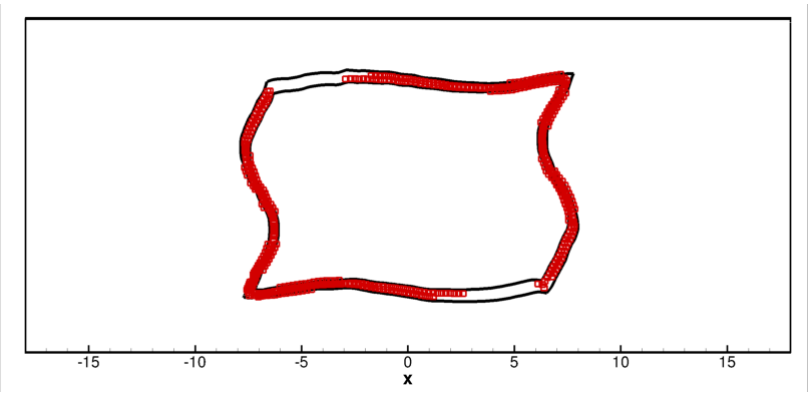


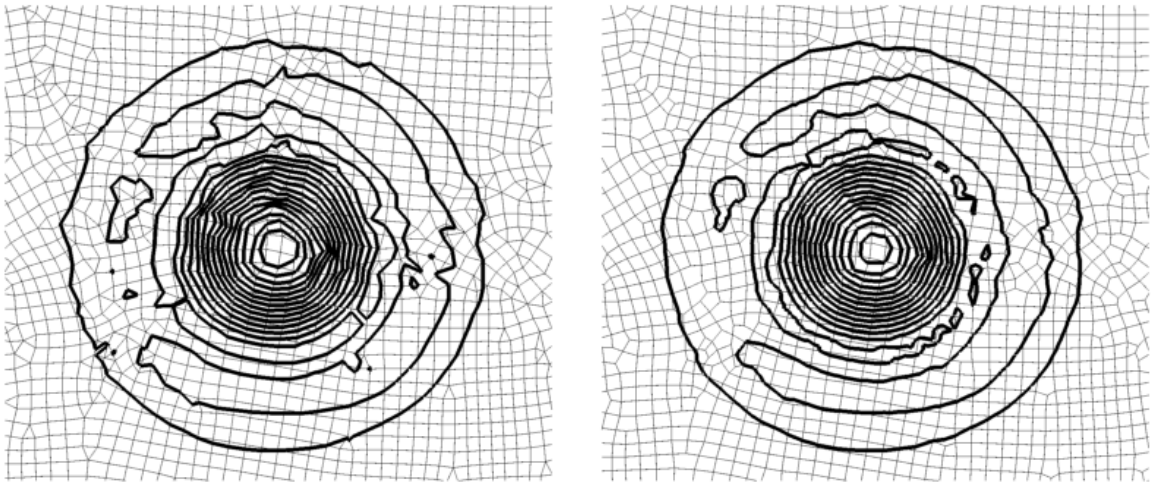
Figure 75: Sample modified overset grid configuration with orphan points shown as red square symbols.

Table 25: Unstructured grids used in the vortex convection study with orphan points

Grids	Grid Spacing (Δs)	Nodes	Fringes	Orphans
Vortex, single grid	0.4	8,528	-	-
	0.36	10,350	-	-
	0.2	33,646	-	-
	0.1	133,640	-	-
Vortex, overset	0.4	11,128	1,154	0
	0.2	40,944	1,894	0
	0.1	162,808	4,074	0
Vortex, overset	0.4	9,352	158	1,040
with orphan points	0.2	33,256	884	1,152
	0.1	129,818	3,394	1,008

Fluctuations in line contours on the patch grid (Fig. 76(a)) may be caused by the inability of the post-processing utility to display smooth contours on mixed-element meshes. To more accurately illustrate the results in this section, solutions on the mixed-element unstructured patch grid have been transferred onto a dual mesh as employed by the flow solver [137]. This is accomplished in an additional post-processing step wherein the nodes of the input mesh are converted to cell centers of a corresponding dual mesh. Polygonal edges are formed by connecting the centers of all surrounding cells. The resulting cell-centered solution representation is smoother

than the original node-centered presentation (Fig. 76(b)), most notably in regions where adjoining cells have different element types.



(a) Input mesh (b) Dual mesh

Figure 76: Comparison of solution visualization on the input mesh and the computational dual mesh.

5.5.3 Results

The numerical experiment described in Section 5.2 has been repeated on the unstructured grid configuration (described by Figs. 74) for cases both with and without orphan points (Table 25). The results presented in this section were computed with the baseline time step of $\Delta t = 0.005$, which was found to be sufficiently small to permit analysis of spatial errors. The effect of temporal errors is discussed in detail in Section 5.5.4.

Baseline Grid ($\Delta = 0.2$)

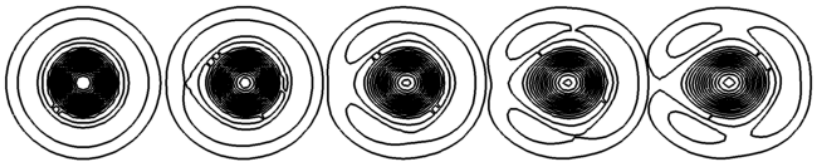
Calculations on the baseline grid resulted in the flow field illustrated by Fig. 77. The orphan-free overset solution preserves the same qualitative features as the single-grid case, which exhibits minor dissipation due to the coarseness of the mesh (Figs. 77a and 77b). If orphan points are present, they are handled by an averaging procedure that

may compromise solution integrity [19]. This averaging dissipates the vortex through the first row of orphans (while entering the overset patch grid) and effectively eliminates the dissipated vortex after the second row of orphans (upon exiting the patch grid), as illustrated by (Fig. 77c). In comparison, the advanced data transfer approach using TPS or W2 interpolation (Fig. 77d) does not result in the uncertainty at orphan locations arising from any averaging procedure. It is able to systematically adapt the data transfer to include more data so that the local solution is still adequately reproduced (Fig. 77d). At the end of the simulation, the vortex strength is comparable to the orphan-free overset case with a slight directional bias in the vortex structure aligned with the overset boundary. The bias occurs because additional donor points in the vicinity of orphans will come from neighboring nodes in a direction tangent to the orphan boundary rather than normal to the orphan boundary. This may be mitigated by further increasing the number of donor points included and sourcing points from farther away in all directions.

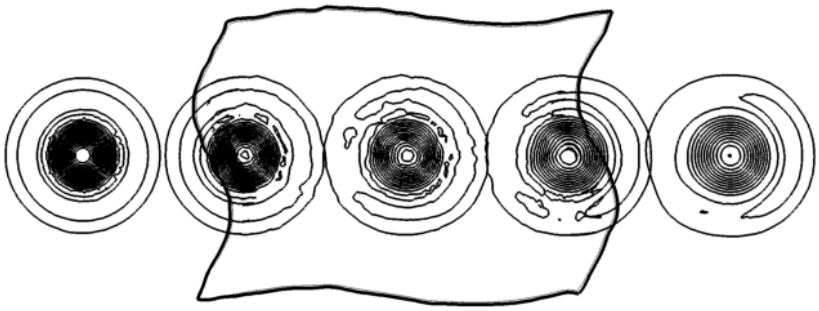
Table 26: Root-mean-square errors for overset solutions with and without orphans compared to the single grid case (baseline mesh, $\Delta s = 0.2$).

Test Case	u error	w error	p error
Single Grid	2.18×10^{-4}	2.70×10^{-4}	3.87×10^{-6}
Overset	2.41×10^{-4}	2.97×10^{-4}	4.36×10^{-6}
Overset, TPS	2.25×10^{-4}	2.87×10^{-4}	4.20×10^{-6}
Overset, W2	2.20×10^{-4}	2.83×10^{-4}	4.14×10^{-6}
Overset with orphans	9.70×10^{-4}	8.92×10^{-4}	7.40×10^{-5}
Overset with orphans, TPS	3.97×10^{-4}	4.90×10^{-4}	6.63×10^{-6}
Overset with orphans, W2	4.01×10^{-4}	4.91×10^{-4}	6.78×10^{-6}

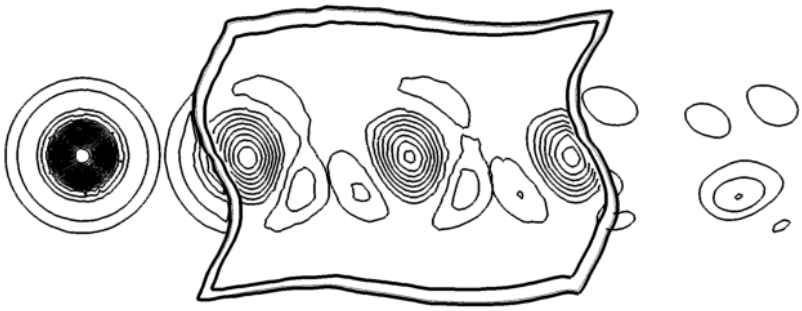
All numerical solutions (single grid and overset, with and without orphans) are compared to the exact solution in Table 26 for the baseline grid case. Since the mesh is relatively coarse, even the single grid case exhibits error due to dissipation. For all three flow variables the overset case without orphans is comparable to the single



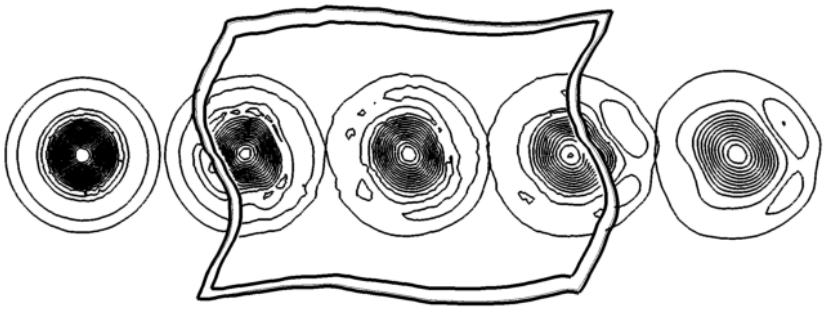
(a) Single Grid



(b) Overset, no orphans



(c) Overset, with orphans and linear mapping



(d) Overset, with orphans and advanced mapping

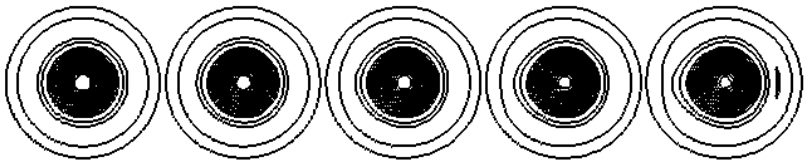
Figure 77: A comparison of the vorticity field for a single-grid case, an overset case without orphans, and two overset cases with orphans. All calculations were made on a baseline mesh ($\Delta s = 0.2$) and five solution snapshots have been super-imposed on top of each other.

grid, with less than 10% root-mean-squared (RMS) error in normal velocity (w) and pressure (p). More significant error accumulation is seen in the streamwise velocity component (u), since the convecting vortex tends to be both dissipated and elongated due to advection (Fig. 77a). This finding is consistent with the conservation error analysis of Section 5.5.5 in which the largest and most persistent errors were observed in the streamwise momentum (Fig. 85). When orphan points are introduced into the computational mesh, the errors in velocity increase by a factor of up to four. The largest change in error occurs in pressure, which increases by an order of magnitude.

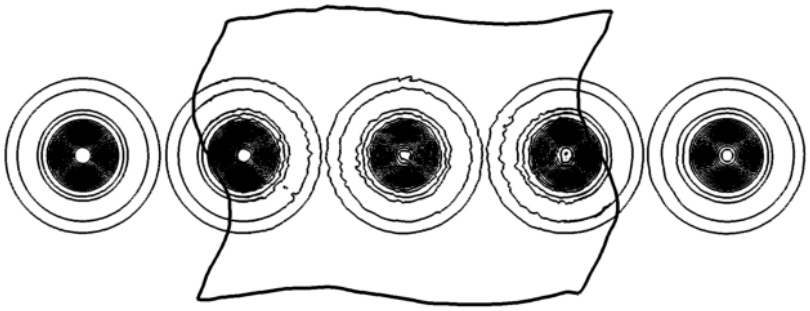
Using an advanced mapping technique (TPS or W2 interpolation) mitigates the RMS error introduced by the gap in the computational domain. The error in streamwise velocity (u), while still higher than the orphan-free overset case, is reduced by about 60%. Similarly, the normal velocity (w) error is reduced by 45%. These values are somewhat higher than the orphan-free case as the interpolated vortex structure becomes skewed as previously noted (Fig. 77d). Finally, the errors in interpolated pressure are reduced by an order of magnitude to the magnitude as the single grid case. Results from applying the W2 basis were virtually identical to the TPS approach, differing and by no more than 2% in all cases. Since the solution fields are visually indistinguishable between TPS and W2, the W2 result has not been presented.

Fine Grid ($\Delta = 0.1$)

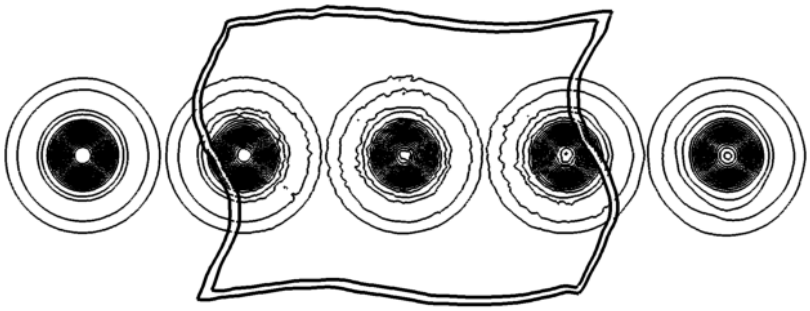
To reduce discretization error and allow more accurate evaluations of overset interpolation error, refined versions of the grids depicted in Fig. 74 (with a grid spacing of $\Delta s = 0.1$) were re-created with reduced overlap regions. Compared to the baseline grid configurations, the vortex is maintained for the duration of the single grid solution (Fig. 78a) with an order of magnitude reduction in velocity errors (Table 27). Errors in the orphan-free overset calculation are again comparable to the single grid



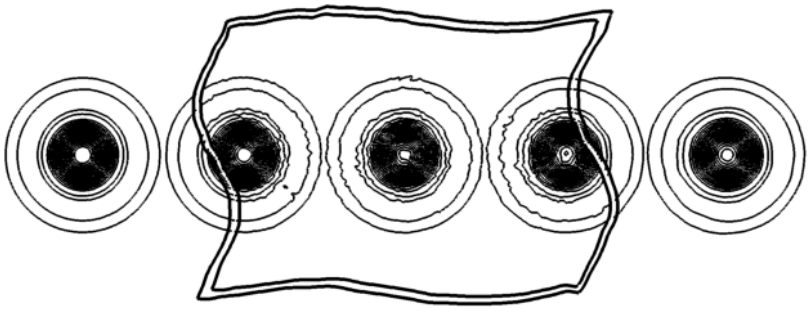
(a) Single Grid



(b) Overset, no orphans



(c) Overset, with orphans and linear mapping



(d) Overset, with orphans and advanced mapping

Figure 78: A comparison of the vorticity field for a single-grid case, an overset case without orphans, and two overset cases with orphans. Calculations were made on a refined mesh ($\Delta s = 0.1$) and five solution snapshots have been super-imposed on top of each other.

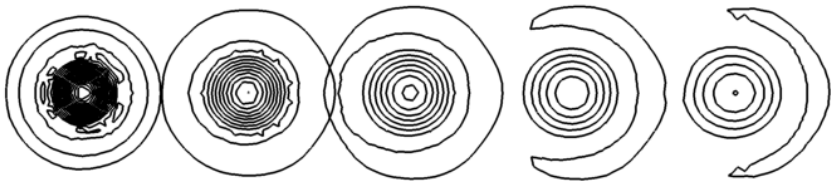
Table 27: Root-mean-square errors for overset solutions with and without orphans compared to the single grid case (fine mesh, $\Delta s = 0.1$).

Test Case	u error	w error	p error
Single Grid	4.94×10^{-5}	7.16×10^{-5}	1.03×10^{-6}
Overset	5.81×10^{-5}	8.02×10^{-5}	1.14×10^{-6}
Overset, TPS	5.35×10^{-5}	7.73×10^{-5}	1.10×10^{-6}
Overset, W2	5.20×10^{-5}	7.61×10^{-5}	1.06×10^{-6}
Overset with orphans	8.61×10^{-5}	1.00×10^{-4}	1.87×10^{-6}
Overset with orphans, TPS	5.95×10^{-5}	8.36×10^{-5}	1.23×10^{-6}
Overset with orphans, W2	5.94×10^{-5}	8.37×10^{-5}	1.27×10^{-6}

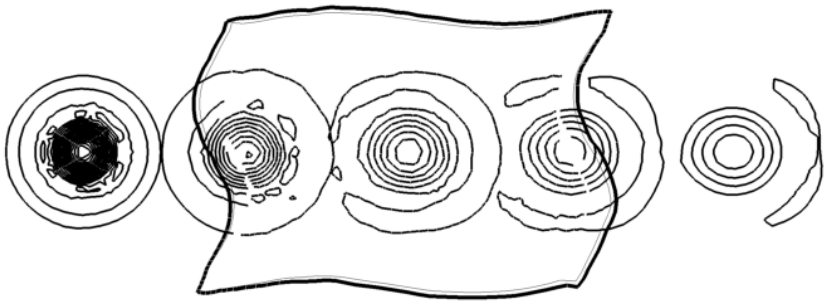
solution (Fig. 78b). The solution with orphan points appears qualitatively similar but has 28–74% higher error compared to the orphan-free solution (Table 27), with the largest error in streamwise velocity. Closer inspection reveals that these errors occur below the vortex core (Fig. 78c). In contrast, the TPS (Fig. 78d) and W2 solutions are both able to smoothly preserve the solution across the grid gap and reduce error. The W2 solution field has again been omitted in Fig. 78 because it is virtually identical to the TPS solution. When applying either advanced data transfer technique, the spatial errors are reduced by 31% in streamwise velocity, 16% in normal velocity, and 32–34% in pressure (Table 27). The TPS and W2 solutions with orphan points are within 8% and 11% respectively of a standard orphan-free solution.

Coarse Grid ($\Delta = 0.4$)

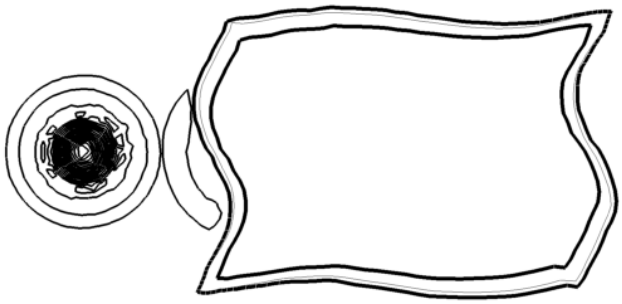
Since grid refinement was demonstrated to effectively eliminate overset interpolation errors, further evaluation of finer grids was not necessary. A coarser grid was studied to evaluate the accumulation of interpolation errors with increasing spatial errors. The cell sizes in this grid configuration are larger than recommended for CFD simulations. The errors in all solution quantities are in general an order of magnitude higher for the coarse grid configuration than the baseline configuration (Table 28). For the trilinear



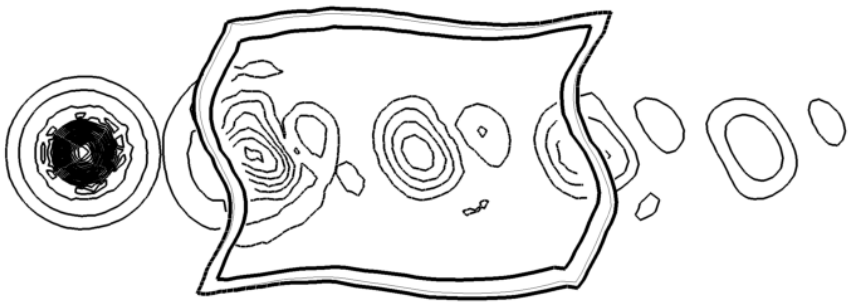
(a) Single Grid



(b) Overset, no orphans



(c) Overset, with orphans and linear mapping



(d) Overset, with orphans and advanced mapping

Figure 79: A comparison of the vorticity field for a single-grid case, an overset case without orphans, and two overset cases with orphans. Calculations were made on a coarse mesh ($\Delta s = 0.4$) and five solution snapshots have been super-imposed on top of each other.

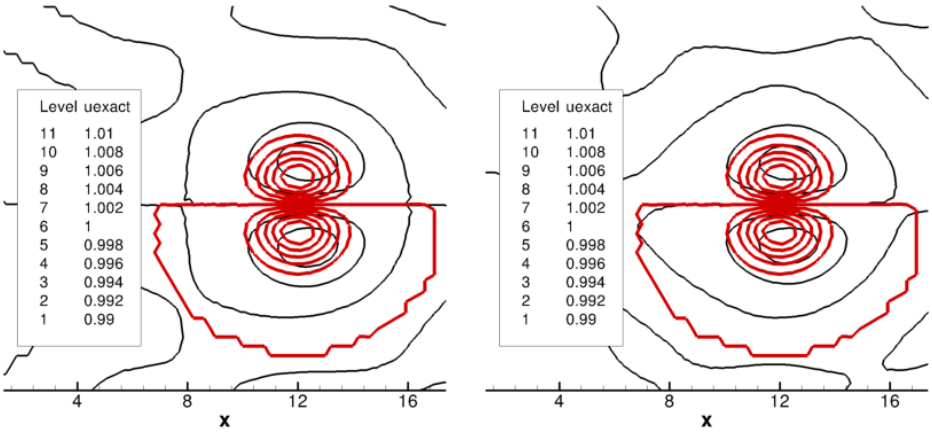
Table 28: Root-mean-square errors for overset solutions with and without orphans compared to the single grid case (coarse mesh, $\Delta s = 0.4$).

Test Case	u error	w error	p error
Single Grid	5.88×10^{-4}	6.42×10^{-4}	1.03×10^{-5}
Single Grid ($\Delta s = 0.36$)	5.33×10^{-4}	5.47×10^{-4}	8.46×10^{-6}
Overset	5.70×10^{-4}	6.45×10^{-4}	1.02×10^{-5}
Overset, TPS	5.43×10^{-4}	6.32×10^{-4}	9.95×10^{-6}
Overset, W2	5.34×10^{-4}	6.29×10^{-4}	1.02×10^{-5}
Overset with orphans	1.05×10^{-3}	1.04×10^{-3}	4.34×10^{-5}
Overset with orphans, TPS	8.82×10^{-4}	9.50×10^{-4}	1.11×10^{-4}
Overset with orphans, W2	8.92×10^{-4}	9.52×10^{-4}	4.27×10^{-4}

overset implementation at an isotropic spacing of $\Delta s = 0.4$, the errors in normal velocity and pressure are nearly the same between the single grid and overset cases, and higher in streamwise velocity for the single grid case. This suggests that errors are dominated by dissipative error rather than overset interpolation error at this level of grid refinement. The $\Delta = 0.4$ single grid configuration had 23% less nodes than the corresponding overset grid configuration where additional grid points arise as a result of the overlapping grid regions. These extra points effectively provide additional resolution for the overset configuration thus reducing errors. To permit comparisons between the single and overset grid configurations, the single grid spacing was slightly reduced from $\Delta s = 0.4$ to 0.36 in order to more closely match the number of nodes (Table 25).

As expected from the test cases on the baseline configuration, the vortex solution is further dissipated on the coarse single grid (Fig. 79a). Figure 80 illustrates the nature of the errors accumulated over the course of the simulation, with elongation and dissipation of the vortex structure. When orphan points are present in the grid, the trilinear overset solution completely dissipates the vortex structure qualitatively. The advanced mapping solution demonstrates improvement over the linear data transfer of

up to 16% error reduction in streamwise and normal velocities, but the effectiveness of the scattered data interpolation is limited by the quality of the sampled data (Fig. 79d). In addition, the error in interpolated pressure p increases with the RBF interpolation methods by up to an order of magnitude.

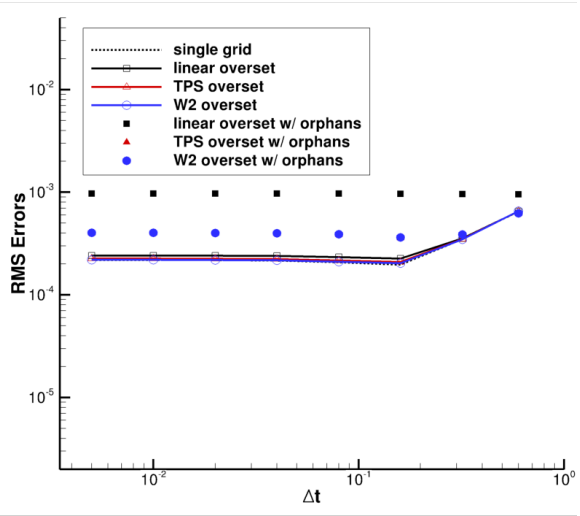


(a) Single Grid (b) Overset, no orphans

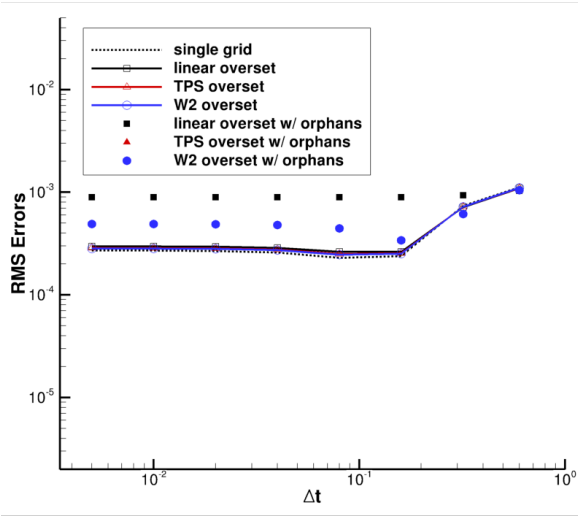
Figure 80: Line contours of streamwise velocity after the final time step. Red contours are the exact solution and the black contours are calculated; the same contour levels are displayed for both cases.

5.5.4 Isolation of Overset Error

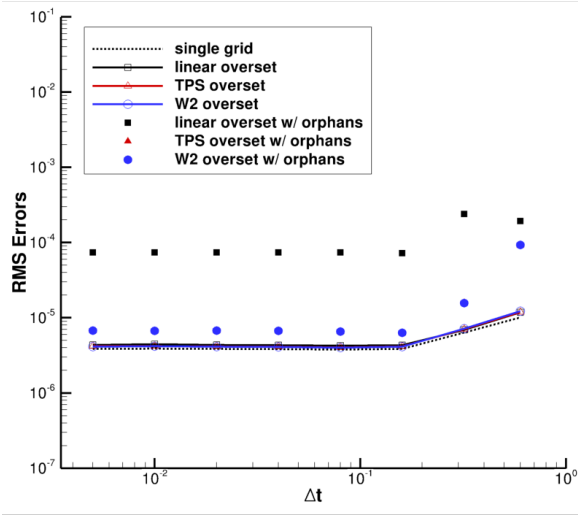
The overall spatial error in an overset solution is sensitive to data transfer strategy. To obtain a true estimate of data transfer effectiveness, the spatial errors should be separated from temporal error. This may be accomplished with high-order temporal schemes and/or small time step sizes so that the theoretical temporal error is negligible compared to the spatial errors. Alternatively, Richardson extrapolation may be used to estimate temporal errors as $\Delta t \to 0$ [138]. A temporal convergence study (with $\Delta t = 0.005$, 0.01, 0.02,..., 0.32) was performed to isolate the spatial error, and series convergence for vanishing Δt was accelerated with the Richardson extrapolation technique. Based on the extrapolated errors, the overset interpolation error has been isolated by subtracting the single grid spatial error from the overset spatial errors.



(a) Streamwise Velocity

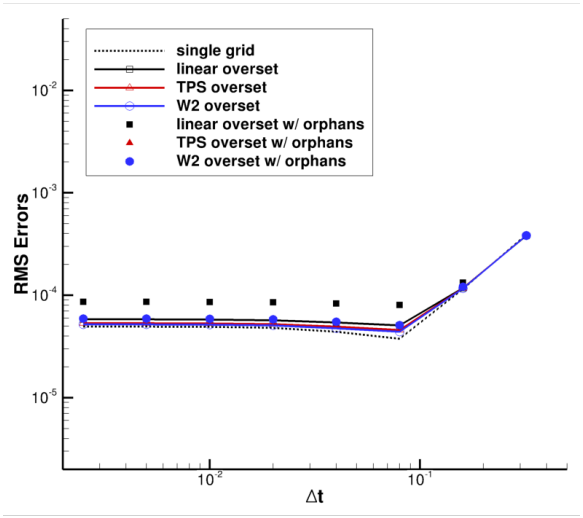


(b) Normal Velocity

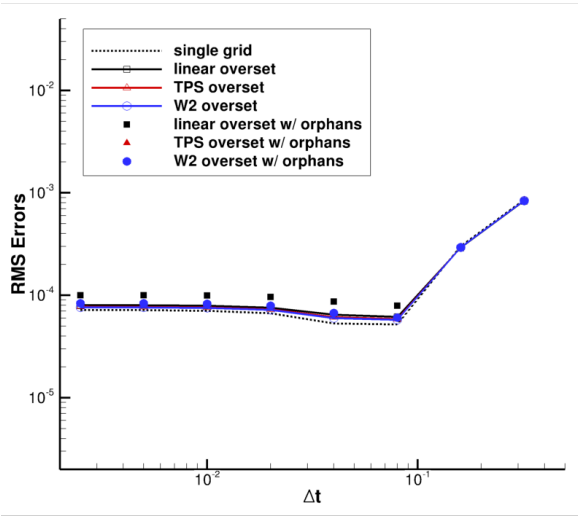


(c) Pressure

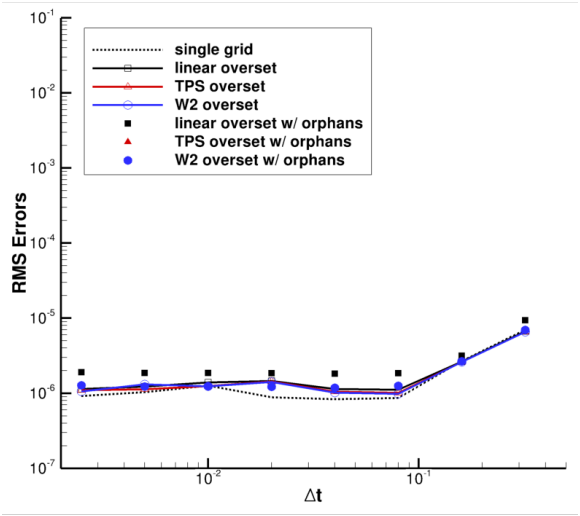
Figure 81: Root-mean-squared error in vortex solution for single and overset grids, with and without orphans, on the baseline mesh ($\Delta s = 0.2$). The single grid curve is on top of the overset curves in all error plots.



(a) Streamwise Velocity

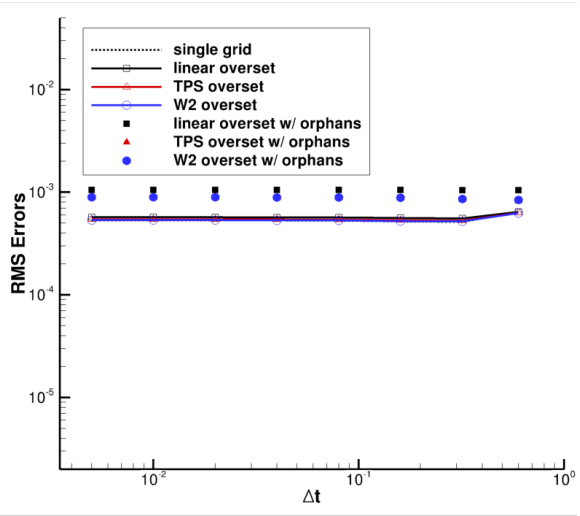


(b) Normal Velocity

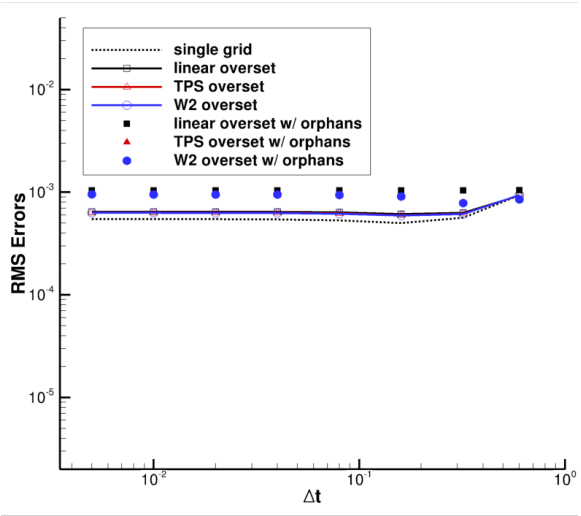


(c) Pressure

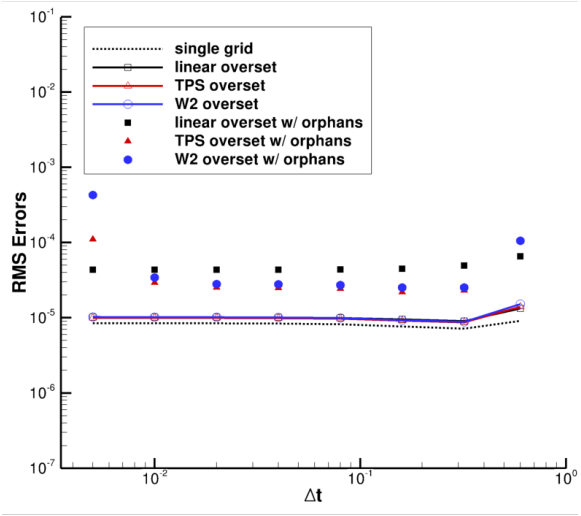
Figure 82: Root-mean-squared error in vortex solution for single and overset grids, with and without orphans, on a fine mesh ($\Delta s = 0.1$).



(a) Streamwise Velocity



(b) Normal Velocity



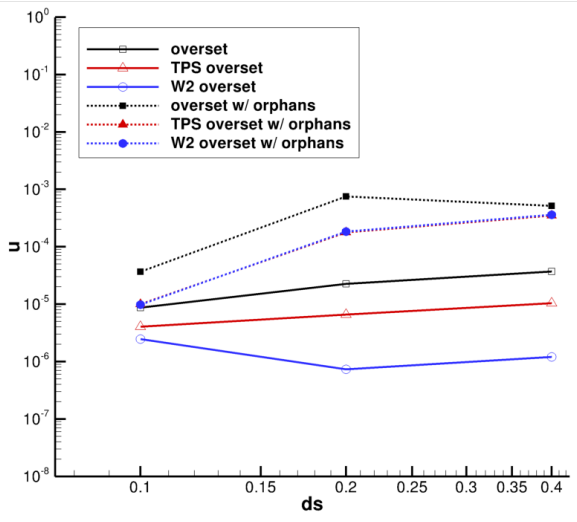
(c) Pressure

Figure 83: Root-mean-squared error in vortex solution for single and overset grids, with and without orphans, on the coarse mesh ($\Delta s = 0.4$). The single grid curve is on top of the overset curves in the streamwise velocity error plot.

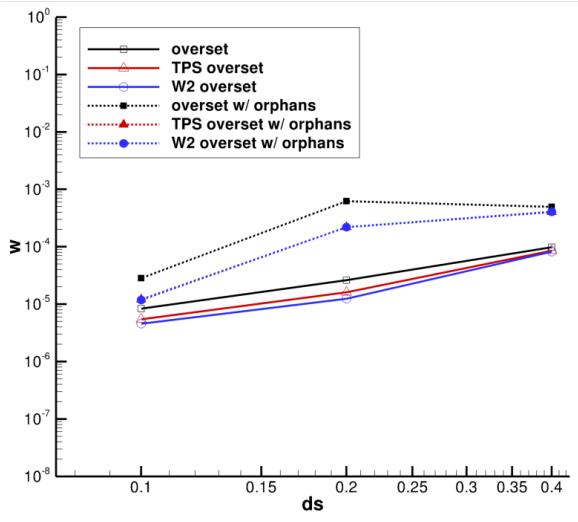
For the coarsest grid configuration tested, the RBF interpolation methods typically resulted in lower errors than the linear approaches (Table 28). An exception was observed in the pressure solution, which only exhibits very small variations by definition due to the C^2 term in Eqn. 74 of the vortex solution definition. Since the magnitude of these variations are small ($\mathcal{O}(10^{-4})$), the pressure field can be more susceptible to interpolation errors. As seen in Fig. 83c, when the time-step size is reduced, the number of iterations (and thus interpolations performed) increases. This increases the accumulation of interpolation errors, which was studied in Section 3.3.2 for a rotor blade solution repeatedly transferred between fine and coarse meshes. The thin-plate spline solution here is more stable than the Wendland C^2 solution, in agreement with the previous findings (illustrated by Fig. 18 in Section 3.3.2). Growth in error begins to appear for $\Delta t = 0.02$ corresponding to 1200 iterations performed. This behavior is not present in the streamwise and normal velocity solutions, and only occurs when the solution is already significantly deteriorated due to both grid coarseness and orphan points. This coarse configuration is an extreme case that would not be applied in practice.

Extrapolation was applied to the RMS error in streamwise velocity (u), normal velocity (w), and pressure (p) and presented in Figs. 81–83. The evolution of spatial error may be understood in terms of the Courant (or CFL) number corresponding to the Courant-Friedrichs-Lewy (CFL) condition, defined as $\frac{V\Delta t}{\Delta s}$. The non-dimensional velocity (V) is normalized by the free-stream velocity in this case, therefore $V = V_\infty = 1$. Values of the Courant number on the order of one are typically sufficient from a numerical stability standpoint although in practice larger values may be acceptable depending on the problem. Since the grid is isotropic, the CFL number depends only on the size of the time step, and when $\Delta t = \Delta s$, the fluid is convected a distance of one cell length with each time step. For the inviscid convecting vortex under investigation, the total error decreases with time-step size and reaches the asymptotic

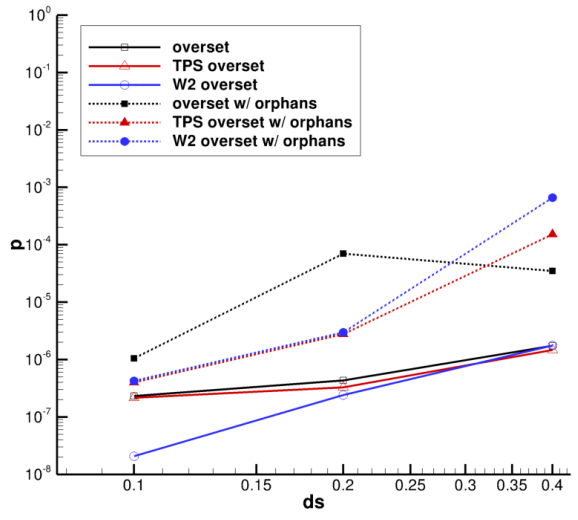
range for $\Delta t \approx \Delta s$, i.e., at a Courant number of approximately one. At smaller time steps, this implies that the temporal error has vanished and the total error is equal to the spatial error. On the baseline grid level, this corresponds to $\Delta t \approx 0.2$ (Fig. 81) while $\Delta t \approx 0.1$ on the fine grid level (Fig. 82). There was a slight increase in error observed as Δt was reduced further due to the number of iterations increasing to maintain a fixed total simulation time.



(a) Streamwise Velocity



(b) Normal Velocity



(c) Pressure

Figure 84: Richardson extrapolated overset errors for various data transfer strategies on configurations with and without orphans.

Table 29: Isolated overset errors on the baseline mesh ($\Delta s = 0.2$), with and without orphans.

Test Case	u	w	p
Linear	2.26×10^{-5}	2.62×10^{-5}	4.34×10^{-7}
Linear, with orphans	7.51×10^{-4}	6.21×10^{-4}	7.02×10^{-5}
TPS	6.58×10^{-6}	1.61×10^{-5}	3.29×10^{-7}
TPS, with orphans	1.79×10^{-4}	2.19×10^{-4}	2.78×10^{-6}
W2	7.33×10^{-7}	1.25×10^{-5}	2.42×10^{-7}
W2, with orphans	1.83×10^{-4}	2.20×10^{-4}	2.97×10^{-6}

Table 30: Isolated overset errors on a fine mesh ($\Delta s = 0.1$), with and without orphans.

Test Case	u	w	p
Linear	8.70×10^{-6}	8.37×10^{-6}	2.32×10^{-7}
Linear, with orphans	3.67×10^{-5}	2.85×10^{-5}	1.05×10^{-6}
TPS	4.04×10^{-6}	5.49×10^{-6}	2.16×10^{-7}
TPS, with orphans	1.00×10^{-5}	1.20×10^{-5}	4.02×10^{-7}
W2	2.47×10^{-6}	4.56×10^{-6}	2.08×10^{-8}
W2, with orphans	9.85×10^{-6}	1.19×10^{-5}	4.25×10^{-7}

Table 31: Isolated overset errors on a coarse mesh ($\Delta s = 0.4$), with and without orphans.

Test Case	u	w	p
Linear	3.72×10^{-5}	9.80×10^{-5}	1.74×10^{-6}
Linear, with orphans	5.17×10^{-4}	4.94×10^{-4}	3.49×10^{-5}
TPS	1.04×10^{-5}	8.55×10^{-5}	1.47×10^{-6}
TPS, with orphans	3.49×10^{-4}	4.03×10^{-4}	1.52×10^{-4}
W2	1.21×10^{-6}	8.19×10^{-5}	1.77×10^{-6}
W2, with orphans	3.60×10^{-4}	4.06×10^{-4}	6.63×10^{-4}

All extrapolated errors are tabulated in Tables 29–31. In general, configurations both with and without orphan points tend to converge at the finest grid level tested for the linear and thin-plate splines overset approaches, and the radial basis interpolation

approaches consistently outperformed linear interpolation. The only exception is the pressure solution on the coarsest grid level, as discussed earlier in this section. Errors were slightly higher at the baseline grid spacing ($\Delta s = 0.2$) than the coarsest grid ($\Delta s = 0.4$) because there were 11% more orphan points in the baseline configuration (Table 25). Typically for configurations with orphan points, the thin-plate splines (TPS) and Wendland C^2 (W2) results were indistinguishable with approximately half an order of magnitude reduction in overset error at the baseline and fine grid levels.

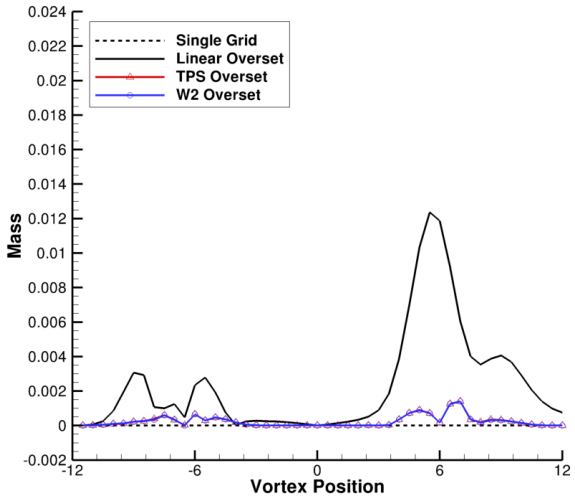
In the orphan-free configurations, the most consistent improvements when applying an advanced mapping were observed in the streamwise velocity, with reductions in overset errors of up to 71% when applying TPS interpolation and over an order of magnitude improvement with W2 interpolation. Improvements of up to 39% and 52% were observed in normal velocity for the TPS and W2 interpolations respectively. However, while the Wendland C^2 function typically had the lowest error, it also displayed the least consistent convergence behavior. For example, overset error in streamwise velocity increased when the grid was refined from $\Delta s = 0.2$ to 0.1; also, the overset error in pressure at $\Delta s = 0.4$ was equal to the linear result.

5.5.5 Conservation Analysis

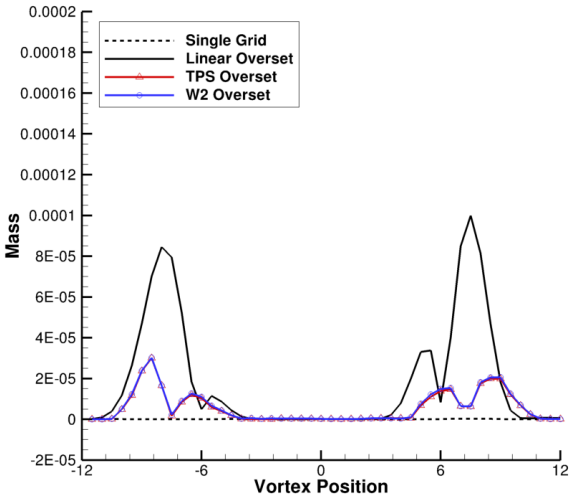
Conservation errors were calculated from instantaneous solutions for both linear and RBF interpolation methods on the baseline and fine grid levels. Mass and momentum fluxes for the entire computational domain were computed along a closed loop formed by the rectangular outer boundary. The extent to which the conservation laws of fluid dynamics are satisfied may be evaluated by considering the net fluxes in and out of the computational domain; any deviation from zero is considered conservation error. Romberg integration was applied to estimate the numerical fluxes in the limit of zero grid spacing. This integration technique approach is based on Richardson

extrapolation and iteratively improves approximations to the exact integral based on successive applications of the trapezoidal integration rule. For $N = 2^k + 1$ equally spaced points where k is a positive integer, the algorithm performs k iterations to remove error terms up to (but not including) $\mathcal{O}(1/N^{2k})$ [14]. To initiate this procedure, the boundary solutions are collocated to provide a number of equally spaced points equal to $2^k + 1$. The shortest integration path (the highest theoretical error) is at the upstream and downstream boundaries, which for the baseline grid ($\Delta s = 0.2$) has 81 points. On these boundaries, the 81 points are interpolated to 129 locations (for $k = 7$) and results in an integration error $\sim \mathcal{O}(\Delta s^{2k}) \approx 10^{-10}$ on the fine grid.

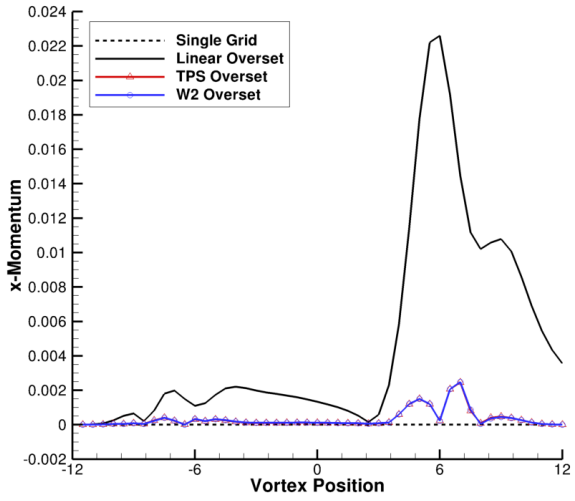
The conservation errors in mass, x-momentum, and z-momentum are compared in Fig. 85 for both the baseline and fine grid levels. For all three quantities, the error (i.e., the net flux) on the single grid is on the order of 10^{-7}, which is greater than the error in the numerical integration, and is therefore significant. Errors are introduced in the overset simulations when the vortex is in the vicinity of the overset interfaces at $x \approx -6$ and $x \approx 6$. These errors appear as local maxima in the computed net flux, with double peaks appearing prominently on the baseline mesh at $x \approx$-9, -5, 6, and 9 (Figs. 85a and e). These peaks correspond to the passage of the front and rear of the vortex core where swirl velocity and velocity gradients are a maximum. After the vortex passage, the initial conservation error level may not be immediately recovered because the maximum number of subiterations was fixed for all simulations. Since the single grid simulation required between 35 and 40 subiterations on average to reduce temporal errors by an order of magnitude, the maximum number of subiterations was set to 40 in all cases. In the overset cases, an order of magnitude reduction in temporal error was achieved in 10–13 steps on average, but the solver residuals remained over an order of magnitude larger than the single grid configuration.



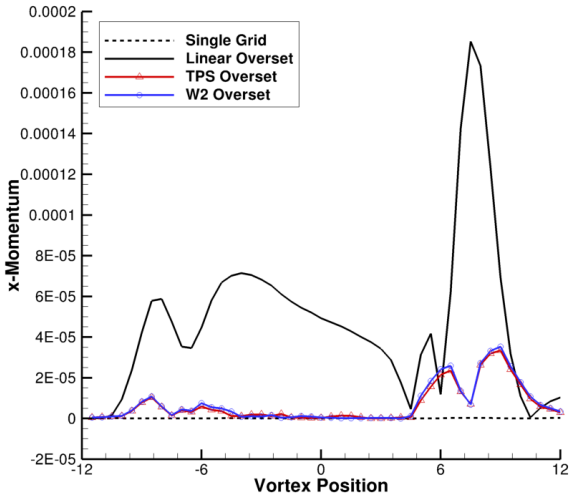
(a) Mass, baseline mesh ($\Delta s = 0.2$)



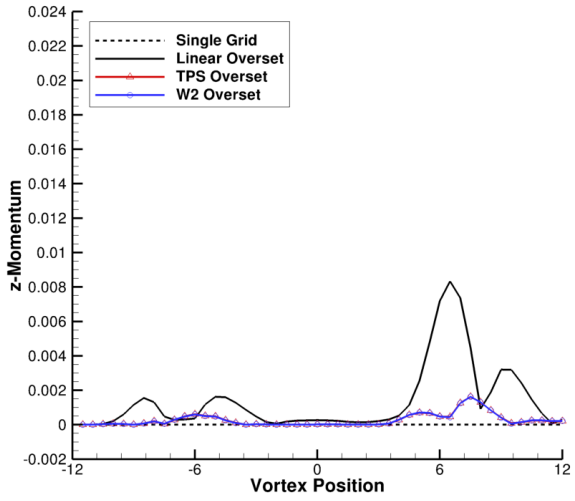
(b) Mass, fine mesh ($\Delta s = 0.1$)



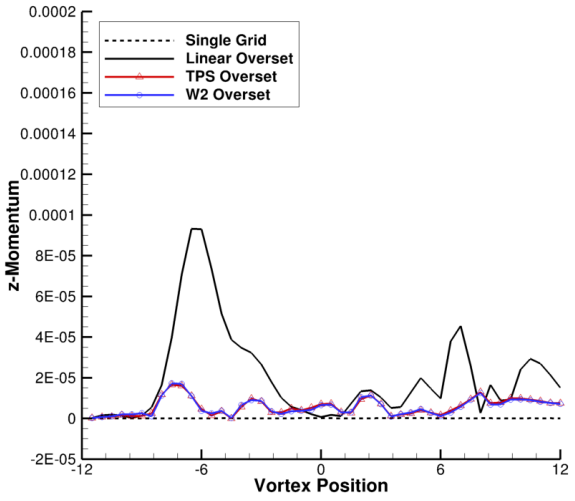
(c) x-Momentum, baseline mesh ($\Delta s = 0.2$)



(d) x-Momentum, fine mesh ($\Delta s = 0.1$)

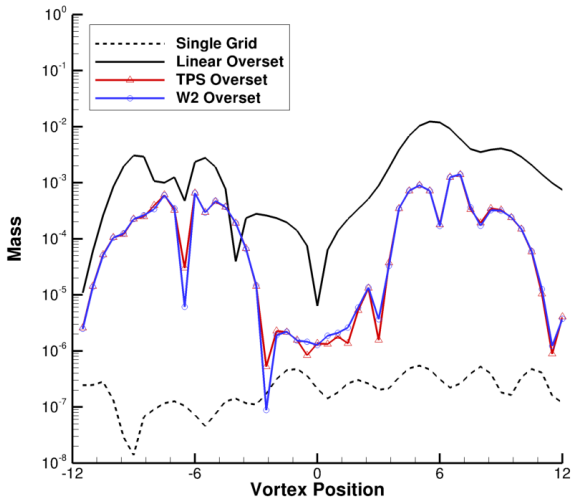


(e) z-Momentum, baseline mesh ($\Delta s = 0.2$)

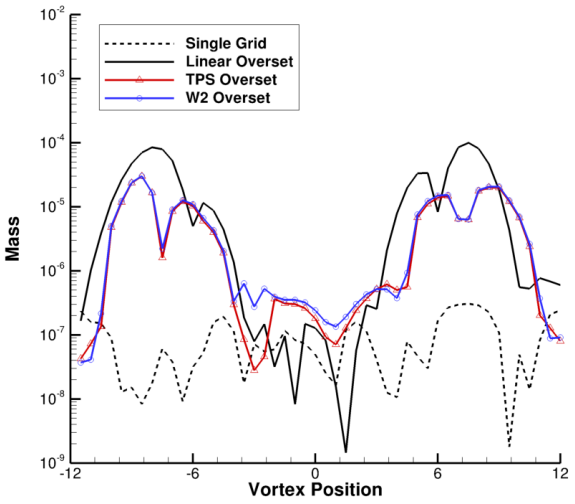


(f) z-Momentum, fine mesh ($\Delta s = 0.1$)

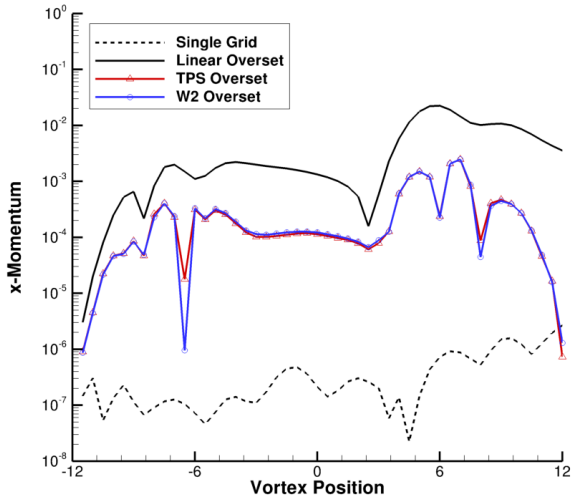
Figure 85: Calculated net flux through the outer boundary of the computational domain for the inviscid convecting vortex test case.



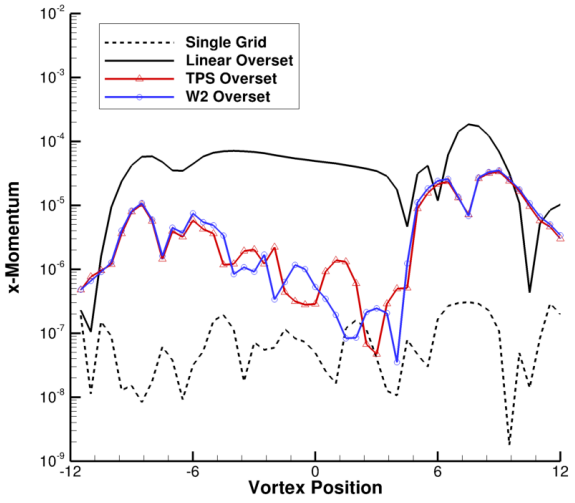
(a) Mass, baseline mesh ($\Delta s = 0.2$)



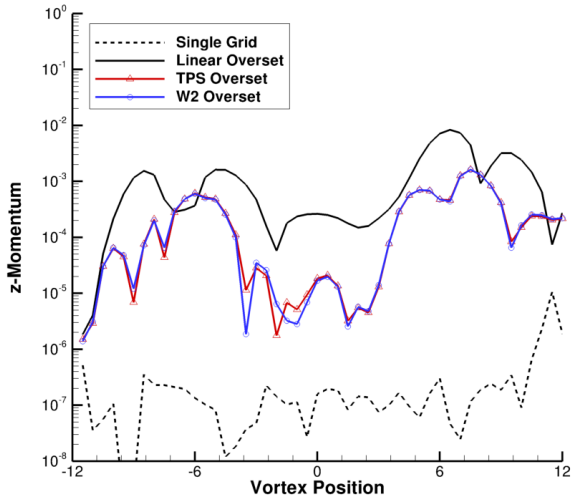
(b) Mass, fine mesh ($\Delta s = 0.1$)



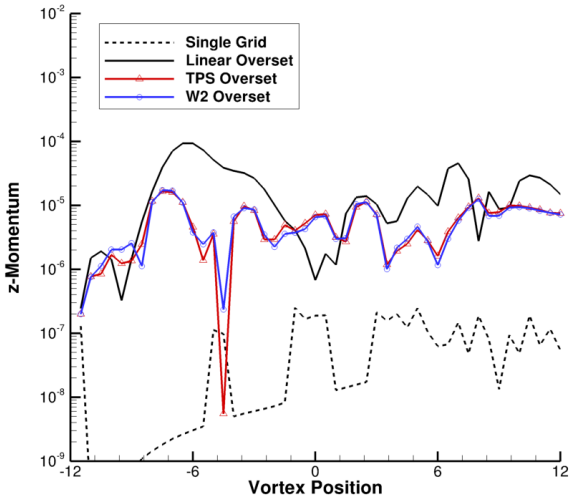
(c) x-Momentum, baseline mesh ($\Delta s = 0.2$)



(d) x-Momentum, fine mesh ($\Delta s = 0.1$)



(e) z-Momentum, baseline mesh ($\Delta s = 0.2$)



(f) z-Momentum, fine mesh ($\Delta s = 0.1$)

Figure 86: Calculated net flux through the outer boundary of the computational domain for the inviscid convecting vortex test case, presented on a semi-log scale, corresponding to the solutions in Fig. 85.

Errors decreased by approximately two orders of magnitude when the vortex control volume was approximated with the finer mesh (Fig. 85). For all conserved quantities at both grid refinement levels, the application of advanced data transfer techniques reduced maximum conservation errors by approximately 80%–90%, nearly an order of magnitude (Tables 32 and 33). For this test case, the thin-plate splines and compactly-supported Wendland C^2 overset solutions were nearly indistinguishable. Of the three conserved quantities, the net mass flux has the most predictable response to the vortex passage, in that the perturbation to the global fluxes caused by overset interpolation error decays quickly and consistently (Figs. 85a and b). Fluxes were plotted on a linear scale to emphasize the effects of vortex passage through overset boundaries. In comparison, the momentum fluxes have more complex responses to the vortex passage due to the nonlinear couplings between momenta in the Euler equations. These are manifested in the persistent errors in x-momentum between $x \approx -6$ and $x \approx 3$ (Figs. 85c and d), as well as the fluctuations in the z-momentum error (visible in Fig. 85f).

At both grid refinement levels, the mass conservation error returns to the approximate unperturbed level at $x \approx -4$ where the vortex core has crossed from the background onto the overset patch grid. With the exception of the z-momentum flux on the fine grid (Fig. 85f), the vortex exiting the patch to the background grid through the second set of fringe points (beginning at $x \approx 4$) has consistently higher error than the passage through the first set of fringe points. Moreover, the large increase in conservation error on the baseline grid is accompanied by a delayed return to the unperturbed level; for example, the mass and x-momentum fluxes have not returned to initial levels even at the end of the simulation (Figs. 85a and c). For more complex aerodynamics, the accumulation of overset conservation errors may be more significant and require additional spatial and temporal refinement than in current engineering practice.

While the fluxes for both linear and advanced overset simulations display the same trends over time (or vortex position), applying an advanced interpolation technique does not simply scale the errors. In general, the RBF overset conservation errors are less than or equal to the linear result, and tend toward the single grid result. For example, in Fig. 86a, as the vortex convects through the overset boundary with orphans, the mass flux errors are within an order of magnitude of the linear calculation. After the vortex has cleared this boundary, the conservation error drops by over two orders of magnitude. This behavior is also observed in the x-momentum (Fig. 86c) and to a lesser degree the z-momentum (Fig. 86e). In general, the reductions in overset conservation error are smaller on the fine mesh since the errors are already lower by about two orders of magnitude. As an exception, the error in x-momentum (which tend to be the highest of the three fluxes) can converge at a higher rate than the linear overset method (Fig. 86d).

Table 32: Maximum conservation errors for a vortex convecting through an overset patch on a coarse grid configuration ($\Delta s = 0.2$).

Overset	Background to Patch			Patch to Background		
Method	Mass	x-Mom.	z-Mom.	Mass	x-Mom.	z-Mom.
Linear	3.06×10^{-3}	2.21×10^{-3}	1.63×10^{-3}	1.24×10^{-2}	2.26×10^{-2}	8.34×10^{-2}
TPS	6.59×10^{-4}	4.06×10^{-4}	6.16×10^{-4}	1.40×10^{-3}	2.46×10^{-3}	1.62×10^{-3}
W2	6.60×10^{-4}	4.08×10^{-4}	6.05×10^{-4}	1.41×10^{-3}	2.48×10^{-3}	1.63×10^{-3}

Table 33: Maximum conservation errors for a vortex convecting through an overset patch on a fine grid configuration ($\Delta s = 0.1$).

Overset	Background to Patch			Patch to Background		
Method	Mass	x-Mom.	z-Mom.	Mass	x-Mom.	z-Mom.
Linear	8.46×10^{-5}	7.15×10^{-5}	9.33×10^{-5}	1.00×10^{-4}	1.85×10^{-4}	4.56×10^{-5}
TPS	3.01×10^{-5}	1.03×10^{-5}	1.66×10^{-5}	1.98×10^{-5}	3.34×10^{-5}	1.31×10^{-5}
W2	3.03×10^{-5}	1.09×10^{-5}	1.73×10^{-5}	2.07×10^{-5}	3.53×10^{-5}	1.26×10^{-5}

5.5.6 Cost

Each of the vortex test cases was simulated on a single processor. In general for the orphan-free configurations, the number of fringe locations at which data transfers were performed was inversely proportional to the isotropic mesh spacing. Reducing the spacing by a factor of two yields an increase in the number of grid points by a factor of approximately four (Table 25). The actual cost increases by a factor of nearly five regardless of the overset method applied (Table 34). This five-fold increase is similar to the findings in Section 5.2.4 regarding preprocessing time. A summary of computational costs is provided in Table 34.

The increase in computational cost for RBF interpolation generally becomes negligible for increasing mesh size, with at minimum 2% increase in computational cost per step at the finest grid level (Table 25). In the presence of orphan points, the increased cost of a RBF interpolation technique over the standard linear approach is more significant, in particular when there is an order of magnitude more orphans than fringe points. The cost of applying the standard overset approach is low in comparison because the standard treatment of orphans with averaging is computationally negligible. For all cases, the cost of applying either radial basis was within 2.5% of each other.

For the configurations without orphan points, accuracy is primarily dictated by grid spacing. For example, the overset interpolation error increases the spatial error by 10–13% with the linear overset method when comparing velocity and pressure errors at the baseline grid level (Table 26). Since the discretization error is reduced at the fine grid level, the increase in spatial error associated with overset interpolation is more significant (11-18%). On the fine grids, application of thin-plate spline interpolation reduced spatial errors by 4–8% in velocities and pressure. Similarly, application of Wendland C^2 interpolation reduced spatial errors by 5–10%. These improvements were realized with less than 3% increase in cost.

Table 34: Cost in terms of averaged wallclock time per step for the test cases and methods presented.

Case	Method	Grid Spacing	Fringes	Avg Subiters.	Cost / step [s]	% change
Vortex	Linear	0.4	1,154	8.08	0.4167	-
Vortex	TPS	0.4	1,154	8.29	0.4860	16.6%
Vortex	W2	0.4	1,154	8.32	0.4789	14.9%
Vortex	Linear	0.2	1,894	8.87	2.1718	-
Vortex	TPS	0.2	1,894	9.02	2.3416	7.8%
Vortex	W2	0.2	1,894	9.02	2.2868	5.3%
Vortex	Linear	0.1	4,074	9.80	10.3831	-
Vortex	TPS	0.1	4,074	9.84	10.6672	2.7%
Vortex	W2	0.1	4,074	9.79	10.6085	2.2%
Vortex w/ orphans	Linear	0.4	158	16.3	0.7830	-
Vortex w/ orphans	TPS	0.4	1,198	39.7	2.1038	168.7%
Vortex w/ orphans	W2	0.4	1,198	39.6	2.1021	168.5%
Vortex w/ orphans	Linear	0.2	884	36.2	4.6648	-
Vortex w/ orphans	TPS	0.2	2,036	37.1	5.2125	11.7%
Vortex w/ orphans	W2	0.2	2,036	36.0	5.1956	11.4%
Vortex w/ orphans	Linear	0.1	3,394	38.6	17.6546	-
Vortex w/ orphans	TPS	0.1	4,402	38.9	20.1944	14.4%
Vortex w/ orphans	W2	0.1	4,402	38.9	20.4251	15.7%

However, when orphan points occur in the grid configuration, the errors introduced in the overset data transfer dominate the solution. In the baseline configuration, streamwise velocity error was quadrupled, normal velocity error was tripled, and pressure error was increased by an order of magnitude (Table 26). Application of radial basis function interpolation reduced streamwise and normal velocity errors by 59% and 45% respectively and reduced pressure error by an order of magnitude for an 11-12% increase in computational cost. On the fine grids, errors due to orphan points introduce additional spatial errors of 48% and 25% in streamwise and normal velocity, and 64% in pressure (Table 27). In this case, application of RBF interpolation is able to reduce pressure error by up to 34% and reduces both velocity errors to within 4% of a standard orphan-free overset solution. This is accomplished with less than 16% increase in cost.

Figure 87 illustrates the increase in computational cost in terms of average wall-clock time per solver iteration. This is plotted as a function of the number of total overset fringe points which includes orphan points. For cases without orphans, the cost increase associated with applying RBF interpolation is negligible, despite an up to 17% increase in cost over trilinear interpolation at the coarsest grid level (Table 34). The increase is more apparent when considering configurations with orphan points. In the overset implementation with trilinear interpolation, the solution at orphan points is determined with a cheaper averaging procedure whereas the RBF approach interpolates all points in the same manner. The cost of RBF approaches, which was insensitive to the basis function applied, increased by approximately a factor of two at the finest grid level (with over 4,000 fringe locations) when orphans were present. In this configuration, the cost increase comes from an approximate four-fold increase in the number of subiterations required to reduce the temporal error by an order of magnitude (Table 34), which is offset by the number of donor points decreasing by between one-fifth to two-fifths (Table 24). The decrease in average donors per fringe

point is due to orphaned points being excluded from the data transfer (which may be an interpolation or extrapolation). As mesh density increases, the number of suitable donor points also increases so that the number of donors approaches the same number as the orphan-free simulations (Table 24), thus further increasing cost.

As the number of fringes increases in an overset problem, the cost of the RBF algorithm grows more rapidly than with trilinear interpolation (Fig. 87). In practice, the number of fringe points typically scales with the overall number of nodes in the grid. Large node-count grids motivate parallel solver execution, which would reduce the number of fringe points per processor. This is expected to minimize any computational bottleneck that high-order interpolation may create.

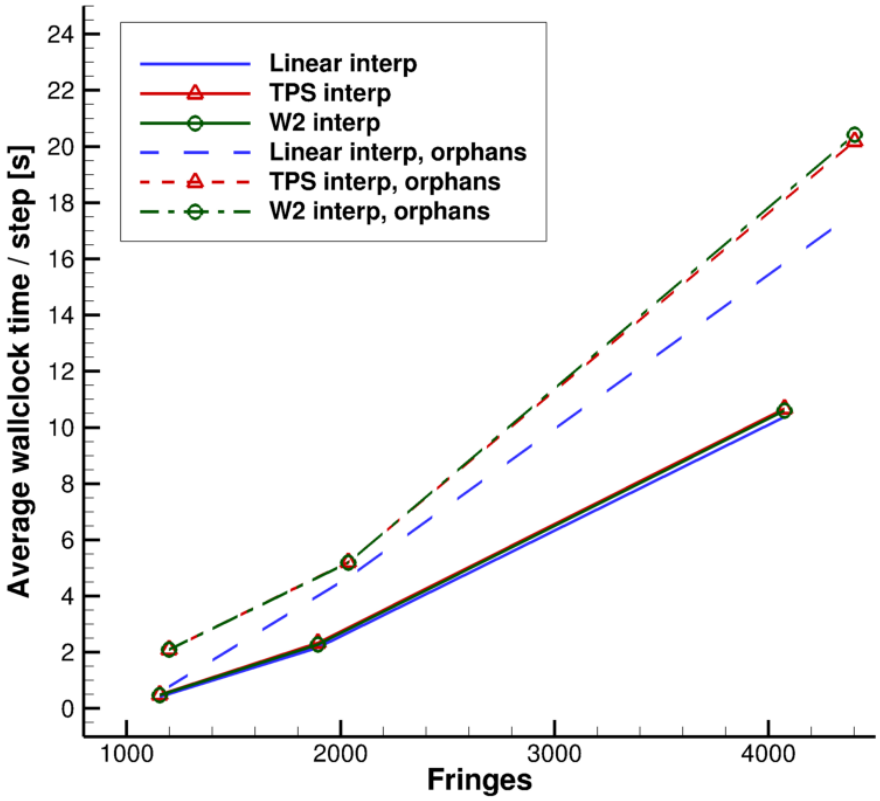


Figure 87: Average wallclock time required per solver iteration as a function of the number of overset fringe points.

CHAPTER VI

HYBRID CASES

In the hybrid approach, the near-body URANS solution performed by FUN3D is coupled to a far field vorticity-velocity (V-V) analysis, VorTran-M. The FUN3D domain is overset on an adaptive Cartesian wake mesh, and hole-cutting is replaced by a direct blanking procedure. The first section describes the interfacing approach between the two solvers. In the following section, basic validation cases will be briefly discussed. Then Section 6.3 describes an analysis of the components of the NREL Phase VI wind turbine, demonstrating hybrid solutions on static and dynamic meshes with various configurations. Finally, the higher-order data transfer methods from Chapter 3 are incorporated into the hybrid interface in Section 6.6.

6.1 General Interface Methodology

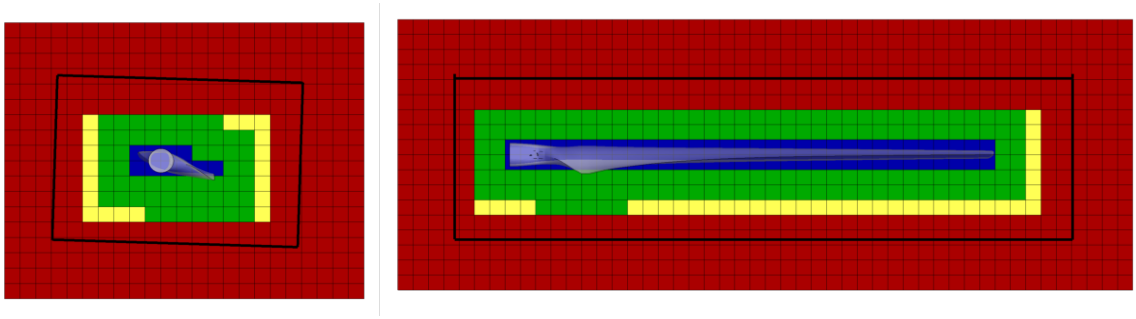
This section describes modifications to the flow solver that permit a hybrid analysis within the framework of FUN3D. First, FUN3D must transfer the near-body solution to the VorTran-M wake domain (Section 6.1.1). Then, the influence of the complete wake calculated by VorTran-M is enforced on the outer boundaries of the FUN3D domain (Section 6.1.2). Arbitrarily overset, non-contiguous meshes may be treated within the hybrid analysis through a decoupled boundaries approach (Section 6.1.3).

6.1.1 Vorticity Insertion

Prior to flow solution, automated preprocessing is required to populate an array of overlap region definitions for the wake solver. These definitions determine the data that are transferred from the near-body domain to the background wake domain after each solver iteration. First, all background cells that contain near-body nodes

belonging to a solid boundary (e.g., a viscous wall or a tangency condition) are marked as level 0 (filled blue contours in Fig. 88). Background cells abutting the level 0 region and outside of solid bodies are marked as level 1 cells (green-colored). Cells that lie outside of level 1 but still overlap with the near-body domain are identified as level 2 (yellow-colored). An optional input parameter permits adjustment of the level 2 extent. All cells outside of level 2 are not guaranteed to be within the near-body domain and are marked as level 3 (red-colored). The objective is to maximize the number of level 1 and 2 cells overlapping with fine regions of the near-body mesh. A buffer region (of at least one layer of level 3 cells where no solution is transferred to the wake) should be maintained between the edge of level 2 (where solution transfer is permitted) and the near-body outer boundary. In practice, some user intervention was necessary to prevent cells adjacent or near to the near-body outer boundary from being marked as level 2, which led to undesirable feedback effects between solvers at the interface boundary. For computational efficiency when dealing with moving grids, the overlap region identification procedure occurs only once when the grids are in their original orientation. To eliminate the need to update the overlap regions at each time step, grid points are transformed back to the original orientation to determine the overlap region to which they belong and whether or not a solution transfer should occur at that location. Velocities are then interpolated at the instantaneous locations to the wake mesh.

To enforce the near-body solution, the vorticity in cell levels 0 through 2 is cleared at each time step. The near-body velocity field lying in overlap region 1 and 2 is then transferred from the URANS solver to the wake solver nodes. The nodes in the level 0 regions that do not abut level 1 regions (i.e., they lie within a solid body) have zero velocity. Wake vorticity is then approximated by an average of finite differences along cell edges (of the wake mesh) for cell levels 0 through 2. Cells that are completely within a solid body will have all zero velocity and thus zero vorticity. Cells that lie on



(a) Axial view (b) Spanwise view

Figure 88: Views of the overlap regions identified by the FUN3D/VorTran-M interface for a rotor blade grid. The level 0–4 regions are indicated by blue, green, yellow, and red contours, and the extent of the near-body grid is outlined in black.

solid boundaries will in general have some nodes interior to the surface that have not been set by the near-body solver. All other cells transfer shed vorticity from the near body to the wake. By default, only cells that have all nodes set by the near-body solver will update the wake solution. However, to rigorously enforce the effect of a bound vortex, the transfer of vorticity from boundary cells (for which some nodes have identically zero velocity) has also been considered in Section 6.4.

Since the near-body URANS nodes (which may be part of a moving and/or deforming mesh) and the wake solver nodes are in general not coincident, data transfer necessitates an interpolation technique. The baseline implementation employs an inverse isoparametric mapping with trilinear basis functions to linearly interpolate within near-body tetrahedral cells, with mapping coordinates directed calculated by barycentric coordinate transformation (Section 3.1.3). Mixed-element meshes are supported by decomposing cells into multiple tetrahedra. For example, five-node pyramidal elements and six-node prismatic elements are decomposed into two and three tetrahedra, respectively. This interface approach searches through all grid elements for the near-body computational cell enclosing each wake node in a vorticity transfer region.

As an alternative to linear interpolation, the higher-order data transfer techniques

presented in Chapter 3 that were applied to the overset investigations in Chapter 5 are also applied here. Radial basis function (RBF) interpolation with two basis functions have been considered: thin-plate splines (TPS) and the compactly supported Wendland C^2 (W2) function. This approach is able to take advantage of the scattered-data capability of the RBF technique. Since enclosing cell information is no longer need to form the interpolant, a search assisted by a k-dimensional (k-d) data partitioning algorithm within FUN3D can rapidly locate the nearest near-body node (rather than the enclosing cell) to each wake node. In lieu of performing additional searches, a list of neighbors to the nearest node is formed to make use of existing connectivity information. To address the situation in which a wake node lies within a solid boundary (which would have caused the enclosing cell search to fail in the original implementation), points that have a nearest node identified as a boundary node will be assumed to lie either on or within a body. As a result, this node will be set with zero velocity. This assumption breaks down within the first viscous layer of grid cells. However, since the nodes are near to the surface, their velocity will be approximately zero. Moreover, considering the relative size of a single layer of viscous cells and the relative coarseness of the wake mesh, this situation very rarely arises and was not observed in the cases studied.

6.1.2 Modified Far Field BC with Riemann Invariants

Explicit boundary conditions are used in the standard far field boundary implementation within FUN3D. Velocities normal to the boundary and sound speed are calculated from two Riemann invariants which are constant along characteristics normal to the far field boundary [139]. These were derived in Section 5.3 and are reproduced here:

$$R^{\pm} = U \pm \frac{2a}{\gamma - 1}. \tag{81}$$

These correspond to incoming (R^-) and outgoing (R^+) waves in one dimension. The resulting boundary normal velocity and speed of sound are:

$$U_{b\perp} = \frac{1}{2}(R^+ + R^-) \qquad \text{and} \qquad a_b = \frac{\gamma - 1}{4}(R^+ - R^-). \tag{82}$$

For subsonic inflow and outflow, the R^+ invariant is calculated from interior conditions (i.e., the solution inside the URANS domain) and the R^- invariant is calculated from exterior conditions outside the computational domain (i.e., far-field or free-stream conditions). For supersonic inflow, both incoming and outgoing characteristics are set by the external conditions. Conversely for supersonic outflow, both incoming and outgoing characteristics are set by the internal conditions.

Using Eqn. 82, the boundary velocities are then:

$$\vec{u}_b = \vec{u}_o + (U_{b\perp} - U_{o\perp})\hat{n} \tag{83}$$

for inflow, and

$$\vec{u}_b = \vec{u}_i + (U_{b\perp} - U_{i\perp})\hat{n} \tag{84}$$

for outflow where $(\,)_b$, $(\,)_i$, and $(\,)_o$ refer to values on the boundary, inside the computational domain, and outside the computational domain.

From the definitions of isentropy ($\frac{p}{\rho^\gamma} = s = \text{constant}$), and speed of sound ($a^2 \equiv \left(\frac{\partial p}{\partial \rho}\right)_s = s\gamma\rho^{\gamma-1}$), entropy is calculated from either exterior or interior variables for inflow and outflow conditions, respectively:

$$s_b = \frac{a^2}{\gamma\rho^{\gamma-1}}. \tag{85}$$

The same entropy equation is manipulated to calculate the density on the boundary:

$$\rho_b = \left(\frac{a_b^2}{\gamma s_b}\right)^{1/(\gamma-1)}, \tag{86}$$

and for an ideal gas, boundary pressure is a function of the speed of sound:

$$p_b = \frac{\rho c^2}{\gamma}. \tag{87}$$

The final element in the solution vector, total energy, is:

$$(\rho e)_b = \frac{p_b}{\gamma - 1} + \rho \frac{u_b^2 + v_b^2 + w_b^2}{2}. \tag{88}$$

To account for the combined influence of the wake (and possibly other near-body grids for the decoupled boundaries approach detailed in Section 6.1.3), the incoming Riemann invariant is modified:

$$R^- = U_o(x, y, z) - \frac{2a_o}{\gamma - 1}, \tag{89}$$

where the external conditions vary in space due to wake induced velocities. In the absence of wake disturbances, $U_o(x, y, z) = U_\infty$ which is consistent with the original implementation. A limitation on the applicability for this approach is that the flow through the interface has been modeled as isentropic. To maintain physical consistency in transonic or supersonic flows, the near-body boundaries should be extended so that any shocks in the flow field do not pass through the hybrid interface.

6.1.3 Decoupled Boundaries

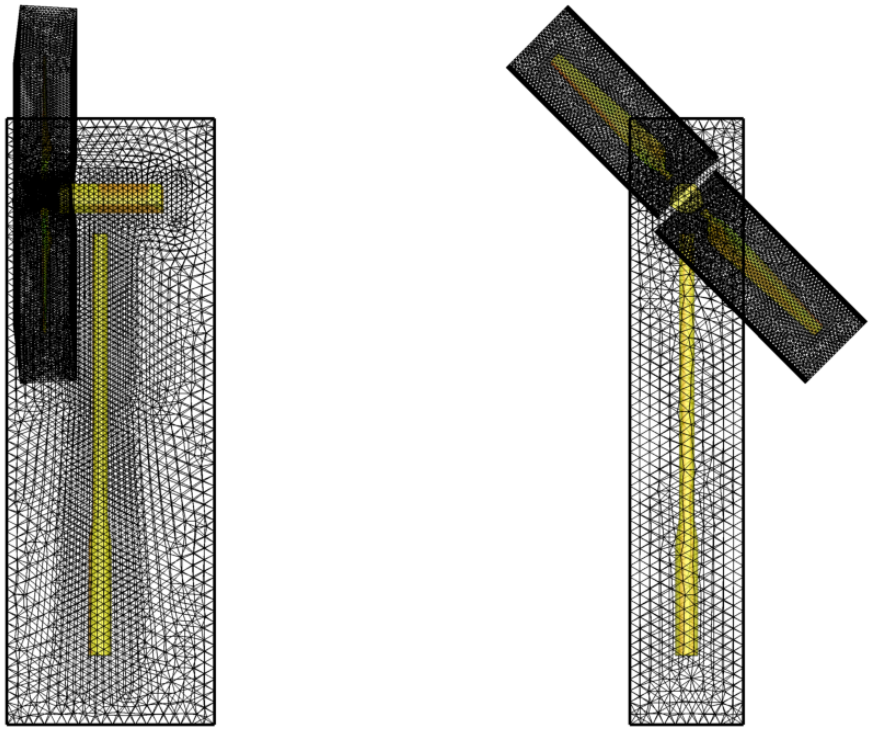
The decoupled boundaries configuration allows for arbitrarily placed grids, both overset and non-contiguous, to be solved within a hybrid framework. With this approach, additional domain connectivity software such as Suggar/DiRTlib is no longer needed. As far as the FUN3D solver is concerned, there exists only a single composite mesh. Solution coupling between grids is accomplished entirely through the VorTran-M wake solver. Solid bodies are still modeled by viscous surfaces in FUN3D to generate vorticity sources for the wake vorticity-velocity analysis. In the case of an Eulerian wake code (e.g., VorTran-M), there is an increase in memory usage because overlap region information (i.e., the locations from which solutions are transferred from FUN3D to the wake) and transfer solutions need to be saved separately for each component mesh. However, these memory increases are modest compared to overall URANS

solver memory requirements because appropriate wake solvers have much less stringent mesh requirements based on both physical and numerical methodology. If a Lagrangian wake code is utilized, there would be minimal additional memory overhead and workload since solution transfer from the near-body solver to the wake comes only from blade loads that are already decoupled from individual component grid boundaries. With either approach, wake calculations and solution transfers from the wake back to the near-body URANS solver follow the existing hybrid implementation discussed in Section 2.3.

Two key differences of this hybrid approach from a standard overset simulation are the complete absence of any background grid (Fig. 89) and the application of hybrid interfaces in lieu of overset boundary conditions. This approach permits non-contiguous grid configurations, simplifying the modeling procedure. Instead of flow field information being interpolated between near-body grids, the solution is transferred to a wake solver which evolves the solution in time and space. In general, the wake solver can provide a continuous mapping of the solution without necessarily using interpolation, and the flow solution can be propagated without interpolation and/or conservation errors. In this manner, the wake solver acts as a more sophisticated connectivity code.

6.2 Validation

To validate the FUN3D coupling with VorTran-M, basic static and dynamic test cases with a NACA0012 wing and finite cylinder (with aspect ratios of 8.8 and 4.0, respectively) were evaluated in a non-rotating frame. Using a free stream Mach number of 0.2 and the incompressible path within FUN3D, the wing was simulated at 8° angle of attack. A 240,000 node tetrahedral mesh extending one chord length beyond the wing in all directions was utilized for these simulations. To provide data for correlation, the two-dimensional lift curve slope (a_{2D}) from Abbott and von



(a) Side view (b) Front view

Figure 89: Sample hybrid configuration with decoupled boundaries for a horizontal-axis wind turbine.

Doenhoff [140] (6.3025/rad) was modified to account for three-dimensional effects with the well-known aspect ratio correction [141]:

$$a_{3D} = \frac{a_{2D}}{1 + \frac{a_{2D}}{\pi AR}}, \tag{90}$$

where a_{3D} is the three-dimensional lift-curve slope, and AR is the aspect ratio. Table 35 illustrates that a notable improvement in the predicted lift is observed when coupled to VorTran-M, using an order of magnitude fewer cells than the FUN3D simulation alone. In addition, VorTran-M is able to effectively model the far field domain, as illustrated by the 5.7% error reduction with the V-V analysis when no background grid is calculated.

Table 35: FUN3D and FUN3D/VorTran-M predicted lift for NACA0012 wing at $\alpha = 8°$, compared with experimental data from [140].

Background grid	**Coupled to V-V**	**Total # of Cells**	C_L	Error (%)
none	no	270k	0.7732	7.9
far field $\rightarrow 5c$	no	4,400k	0.7616	6.3
none	yes	420k	0.7326	2.2
-	-	-	0.7166	0.0

A dynamic case was then evaluated with the wing pitching at $\alpha = 8 \pm 5°$ with a reduced frequency of $k = \omega c/2U_\infty = 0.5$. The simulation successfully demonstrates dynamic update of the overset insertion region in which the flow field is passed from FUN3D to VorTran-M (Fig. 90) from a rotating grid. The VorTran-M domain is dynamically resized to encompass the convected vorticity that remains purely outside the FUN3D domain. Preservation of shed vorticity including the starting vortex was observed and the VorTran-M solution predicts three-dimensional wake development (Fig. 91). The difference between the traditional URANS and hybrid methods on the unsteady airloads is negligible in this case, demonstrating that solution fidelity is preserved for the hybrid method (Fig. 92). In comparison to theory, the lift shows excellent correlation, matching both magnitude and phase, while the moment shows

a 12% magnitude reduction and 20° phase delay due to the increased wake influence (Fig. 92).

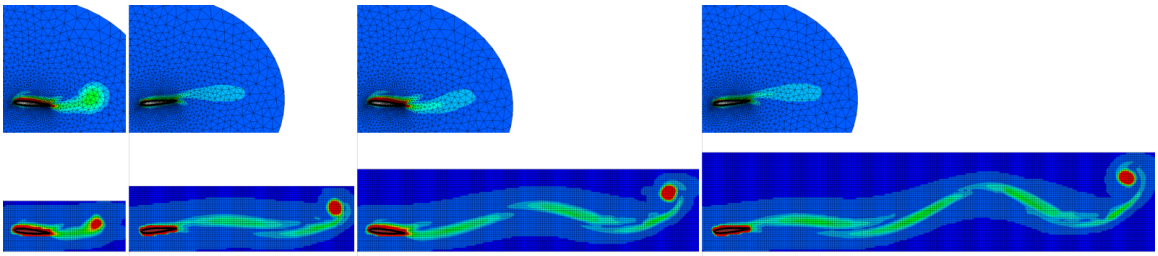


Figure 90: Vorticity magnitude of a pitching NACA0012 wing with FUN3D/VorTran-M coupling, showing the coupled wake evolution after 1-4 vortex shedding cycles. The top row shows the flow field output from FUN3D and the bottom row shows the corresponding flow field from VorTran-M.

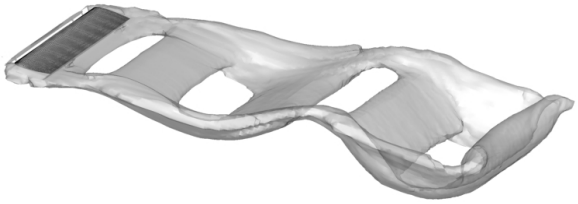


Figure 91: Iso-surface of vorticity magnitude for pitching NACA0012 wing corresponding to the last frame of Fig. 90.

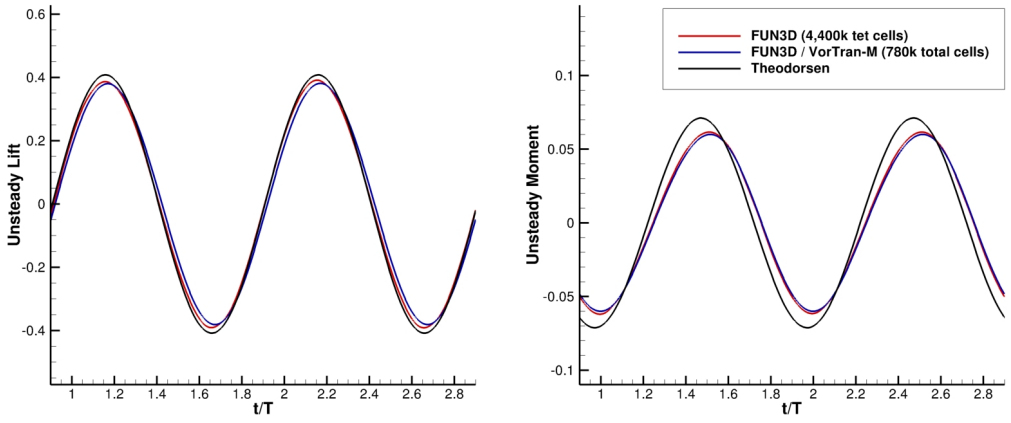
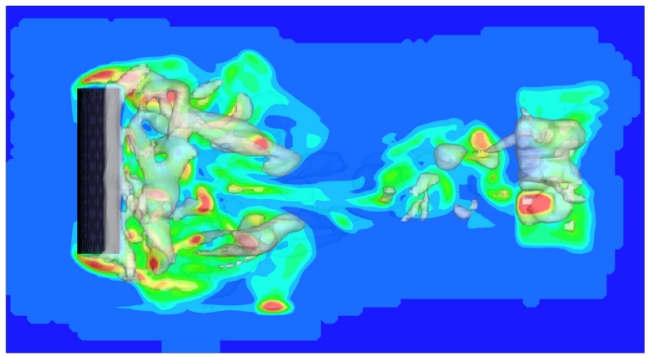
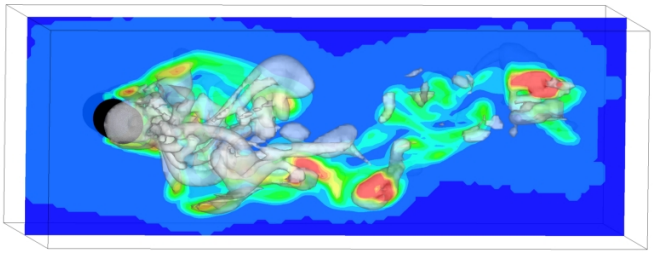


Figure 92: Unsteady lift and moment on the pitching NACA0012 wing.



(a) Overhead view



(b) Side view

Figure 93: Iso-surfaces of vorticity magnitude for finite cylinder in crossflow, with vorticity contours shown in the mid-span plane.

6.2.1 Cylinder in Cross-Flow

The classical problem of a circular cylinder in a cross-flow was evaluated both in terms of predicted Strouhal number and the wake structure. A free-stream Mach number of 0.2 with a Reynolds number of 3900 was investigated. The fully tetrahedral grid consisted of 3 million nodes, which is larger than the prior grids, but which acted as a verification of the capability of the parallel hybrid computations. It should be noted that Lynch [142] found that the tetrahedral grid was not the best for this problem; a mixed-element mesh with specified boundary layer aspect ratio cell sizes and growth was required to capture the most accurate surface characteristics. The FUN3D-alone grid spanned 19.5 diameters downstream, while the near-body FUN3D/VorTran-M grid spanned 3.5 diameters in the wake. The predicted primary Strouhal number for the FUN3D/VorTran-M simulation was extracted at the mid-span and computed to be 0.20, matching the experimental and FUN3D simulation predictions [142] on

the full grid. When running the near-body grid with coupling, significant three-dimensional flows are observed (Fig. 93), as is expected from the configuration.

6.2.2 Rotating Configurations

Demonstration of the methodology for rotating configurations was initially evaluated for the two-bladed hovering rotor of Caradonna and Tung [143, 144] which provides an excellent correlation case with which to evaluate the FUN3D/VorTran-M methodology. The experiments included numerous blade pressure and tip vortex geometry measurements, and have shown that both the blade loadings and vortex trajectories are relatively insensitive to rotor tip speed [143]. For this study, the rotor was run at 1250 RPM ($M_{tip} = 0.439$) with a fixed collective pitch of 8°.

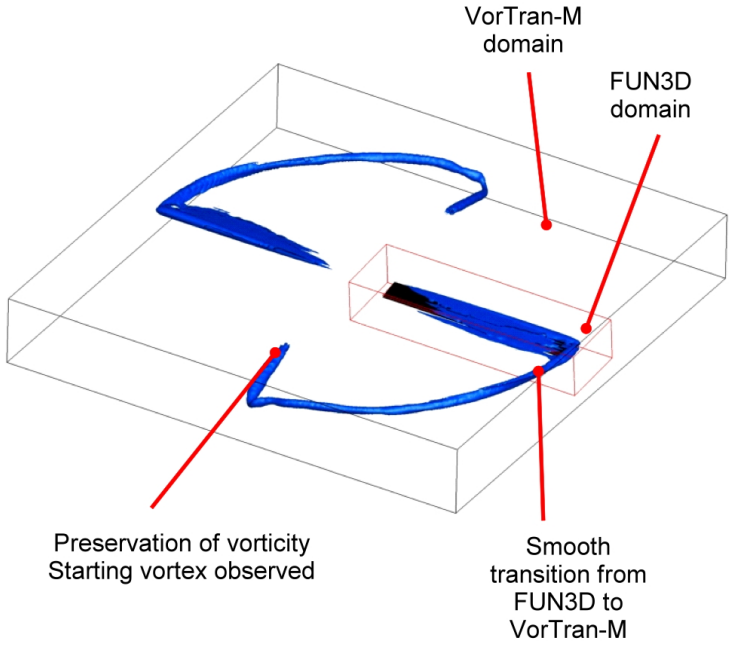


Figure 94: Schematic of FUN3D/VorTran-M grid arrangement for the two bladed rotor in hover.

The rotor was simulated via an overset grid arrangement where only a single blade was directly modeled in FUN3D. Given the axisymmetry of the hover scenario, the predicted flow field that initializes the VorTran-M vorticity distribution is duplicated, rotated by 180 degrees, and inserted as the second blade (Fig. 94). The advantage of

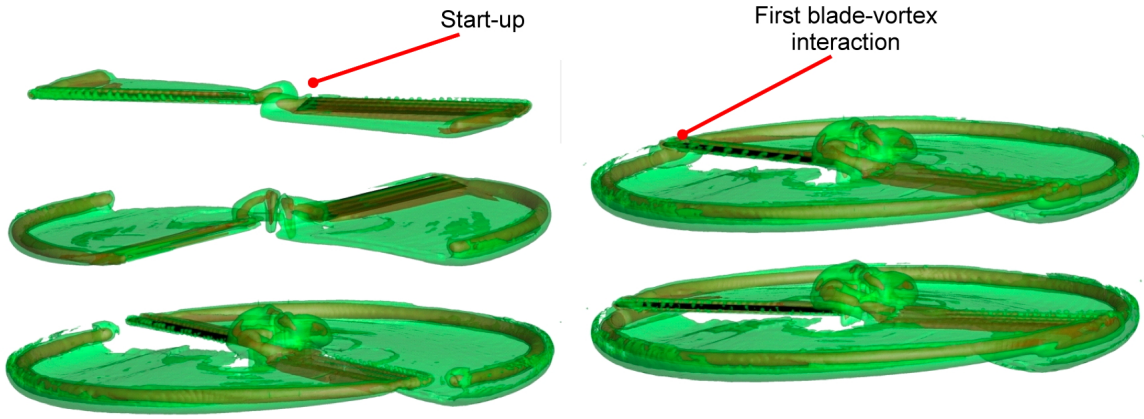
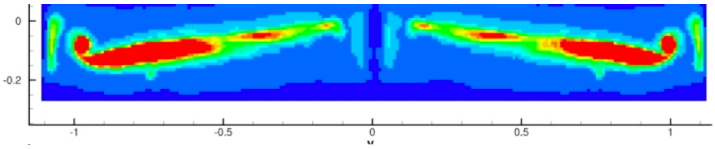
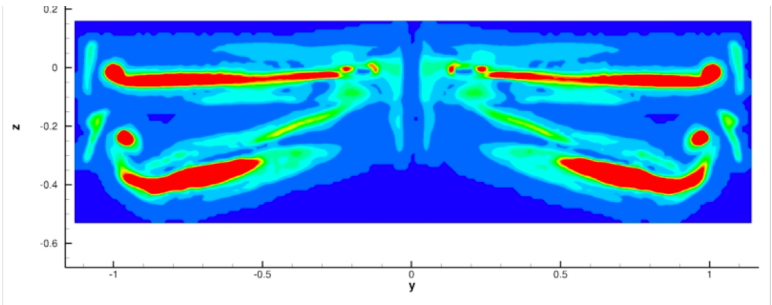


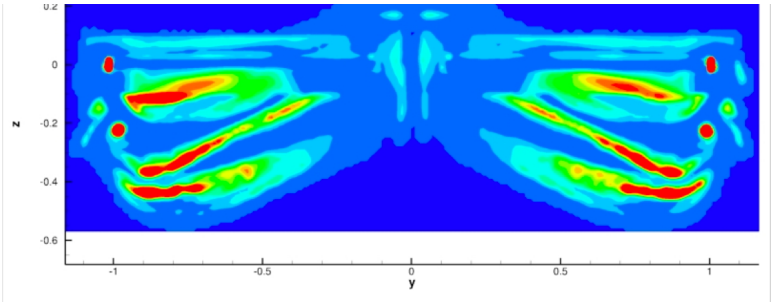
Figure 95: Snapshots of the hovering rotor wake predicted by FUN3D/VorTran-M.



(a) 150°



(b) 300°



(c) 450°

Figure 96: Illustration of the temporally developing wake of a hovering rotor (wind turbine at zero yaw) captured by FUN3D/VorTran-M. The vorticity magnitude illustrates the crispness of the vortex sheet and tip vortex as it moves away from the rotor blade.

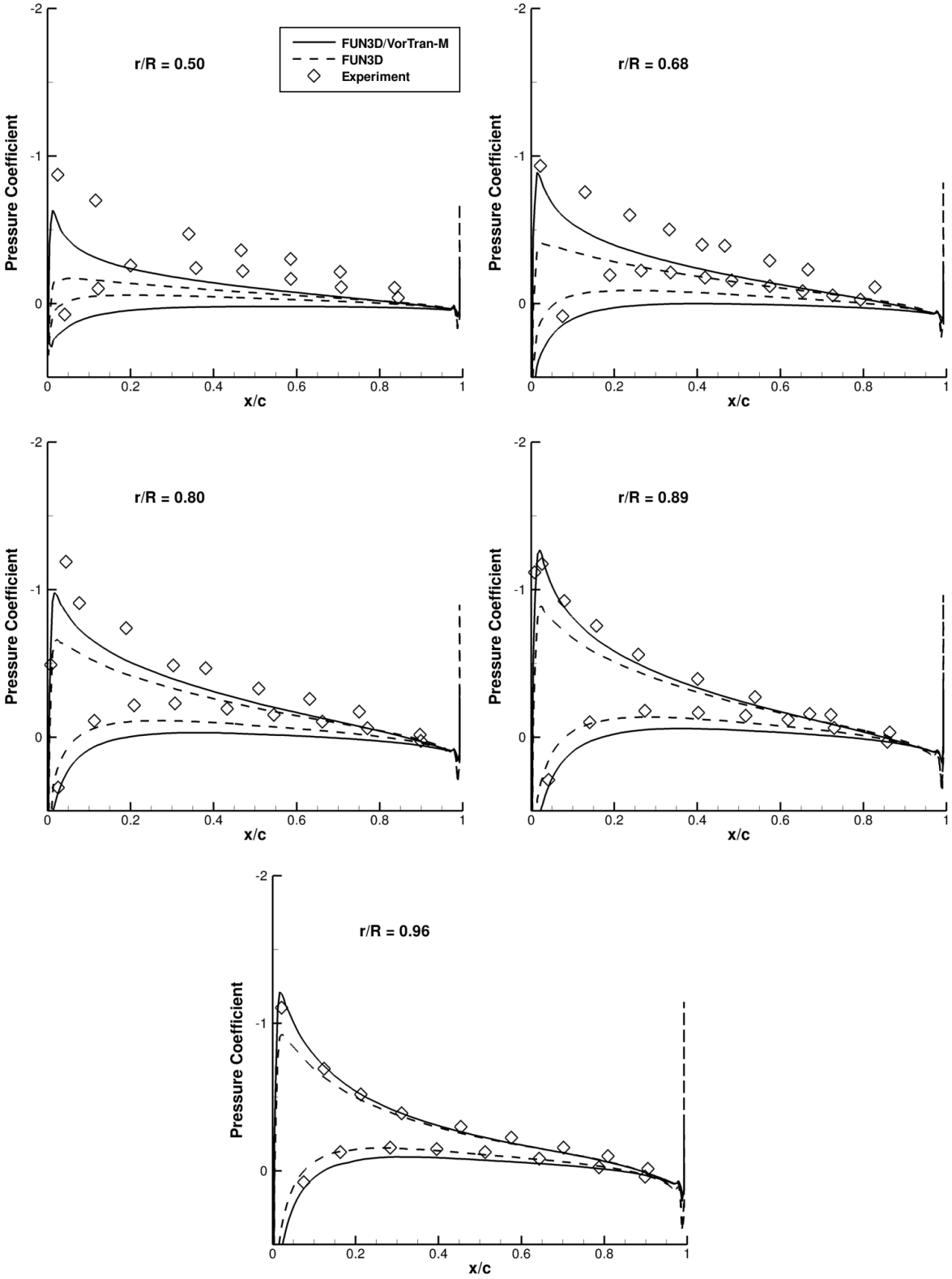


Figure 97: Pressure distributions for the Caradonna-Tung rotor at various radial stations.

this approach is that the cost of the calculation can be reduced through the reduction of mesh nodes (as only one blade is modeled), while still retaining the full unsteady influence of two blades on the rotor wake. A series of snapshots of the rotor wake (Fig. 95) illustrates the capture of the blade-vortex interaction that is present, and demonstrates the build-up of the wake structure as VorTran-M captures the long-age wake in hover. The characteristic of the resulting flow field can be observed via vorticity in Fig. 96 after two and a half partial revolutions. To verify convergence, four full rotor revolutions were simulated with the hybrid methodology and five revolutions were simulated with standalone FUN3D. Sectional load data presented in Fig. 98 indicate that the hybrid solution changes by a maximum of 3.7% between the final two revolutions, and the standalone FUN3D solution changes by a maximum of 5.5%.

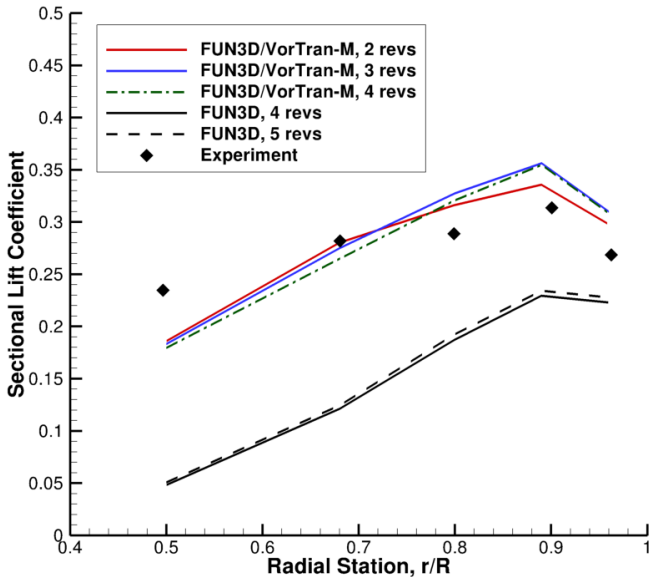


Figure 98: Blade span loading for the Caradonna-Tung rotor.

Sectional lift (averaged over one rotor partial-revolution) compares favorably with experiment (Fig. 98). At the experimental pitch setting, the errors at the four outboard stations range between 6 and 14% for the hybrid simulation, compared to errors between 16 and 56% in the standalone FUN3D case. The lift distributions

up to $r/R = 0.89$ are qualitatively similar, with slopes of 0.48 and 0.45 for the hybrid and standalone cases, respectively. At the blade tip however, the hybrid methodology is able to more accurately capture the lift reduction, matching the slope of the experimental data (-0.66 and -0.73 for the hybrid and experimental results, respectively).

Lift is slightly under predicted on the inboard portion of the blade, as a result of vortices persisting near the blade root. Similar flow features were also observed in calculations performed by other researchers using the structured CFD solver, OVERFLOW [39]. Corresponding pressure distributions at select radial stations are depicted inFig. 97 and demonstrate the same reduction in inboard blade loading. In all cases, the coupled result predicts pressures closer to the experimental values than conventional overset FUN3D predictions. Significant improvements are observed near the tip using FUN3D/VorTran-M coupling, with the experimental location and amplitude of the suction peak being accurately captured at the two furthest outboard stations.

The grid used in the FUN3D/VorTran-M coupling contained 2.2 million nodes, reduced from 5.3 million nodes in the standalone FUN3D overset case. Additional improvement may be obtained with the feature-based adaptation demonstrated in Ref. 145 to focus grid nodes where they are needed for both applications of FUN3D. It is anticipated that, in order for FUN3D alone to produce loads of the same accuracy as the coupled case, significant grid refinement in the wake region will be necessary, thus corresponding to increased computational expense.

6.3 Wind Turbine Components Analysis

6.3.1 Background

The United States has an estimated total wind resource of over 8,000 gigawatts, and wind energy has been forecasted to meet roughly 20% of the national energy

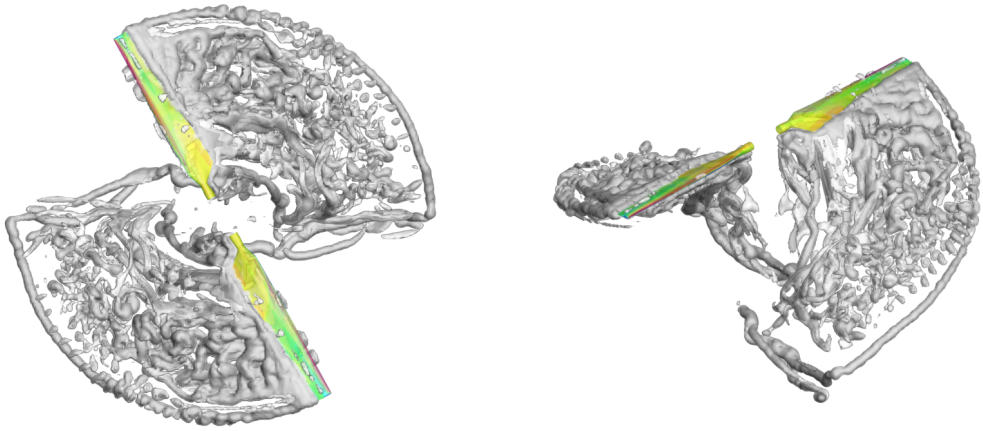


Figure 99: FUN3D/VorTran-M near-body flow field.

demand by the year 2030 [146]. This scenario may be realized by harnessing offshore resources where higher velocity winds are available and energy is harnessed near to regions of the highest demand, thus minimizing transmission cost. A number of key technological improvements which will improve both on- and off-shore power production are summarized in Ref. 146. These include taller towers, larger rotors, and advanced materials, all of which require more sophisticated and cost-effective analysis.

At the individual turbine scale, momentum and energy are transported between the micro-scale and the larger eddies that form in the wake of turbine components. Complex turbulent structures are observed in the wake even when studying an isolated rotor (Fig. 99). The vortical "worm" structures are resolved with an advanced turbulence model that permits capture rather than modeling of the larger turbulent scales, and these results are comparable to simulations performed by other researchers [7]. Furthermore, the flow around bluff bodies such as the turbine tower and nacelle is also expected to be separated and turbulent which increases the physical complexity in the problem.

The turbine wake will expand downstream of the rotor due to the slowed flow,

resulting in the formation and eventual break-up of free-shear layers in the wake. On the wind-farm scale, the wake vorticity from upstream turbines will likely impinge on downstream turbines for most wind farm configurations, causing a build-up of turbulence on the downstream rotor disc; these vortex interactions can significantly alter the flow field, decreasing power output and increasing fatigue loads [147]. Since free air is not free from disturbances, atmospheric turbulence can also play a key role in altering the turbine inflow and consequently the power output.

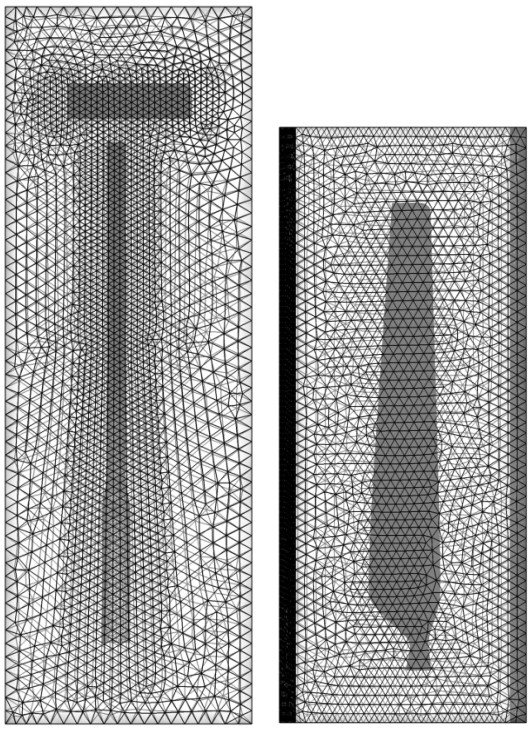
6.3.2 Experimental Data Set

The National Renewable Energy Laboratory (NREL) conducted a series of Unsteady Aerodynamics Experiments, culminating in the Phase VI test of a full-scale 20 kW wind turbine in the 80 by 120-foot wind tunnel at NASA Ames [148]. The wind turbine rotor is 10 m in diameter with two blades built from the NREL S809 airfoil, and is mounted on a nacelle atop an 11.5 m tower (pictured in Fig. 100). While the test turbine is at a smaller scale than utility-scale turbines currently in use (typically on the order of megawatts output), the experiment demonstrated a range of three-dimensional unsteady aerodynamic phenomena. A range of velocities (from 5 to 25 m/s) and a range of yaw angles (0° to 30°) were tested at a nominal rotor speed of 72 RPM for both upwind and downwind configurations. Since a large amount of data from the experiment campaign is available for correlation, a significant portion of the work in this chapter will be devoted to the study of the NREL VI turbine.

The original configuration was simulated with a background mesh extending 8 rotor diameters upstream, 10 diameters downstream, and 5 diameters in each lateral direction Fig. 102. The final mesh size for a two-bladed rotor had a total of 14.8M nodes. In comparison, the hybrid rotor analysis assuming rotational symmetry used only a single rotor blade for a 70% reduction of near-body computational nodes in the URANS solver. For more general scenarios such as modeling yaw conditions or

Figure 100: The NREL Phase VI turbine mounted in the $80' \times 120'$ tunnel at NASA Ames [148]

the full turbine, a decoupled-boundaries approach could still achieve a greater than 30% reduction in node count or over 40% reduction in cell count (Table 36). The cost of the wake solution is in general negligible compared to the near-body solution due to an efficient Cartesian mesh topology and a drastic reduction in cell count, typically $\mathcal{O}(10^5)$ cells solved. Actual observed computational costs are presented in Section 6.5.4. The model does not include the rotor hub, which in the actual experiment included an instrumentation package mounted on a boom extending upstream. This geometry has also been neglected by other researchers [149, 7] and is expected to have a minimal impact on blade loads due to low resultant velocities on the root sections compared to the rest of the blade.



(a) Tower & Nacelle (b) Rotor Blade

Figure 101: Near-body CFD grids representing the NREL VI wind turbine.

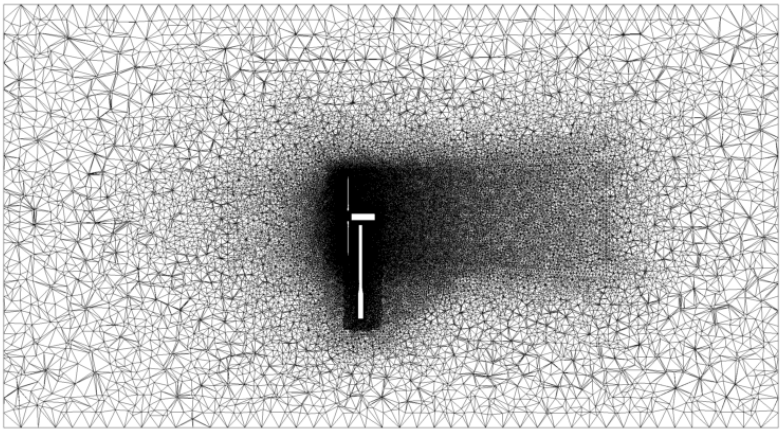


Figure 102: Full CFD configuration for the NREL Phase VI turbine (from Ref. 7)

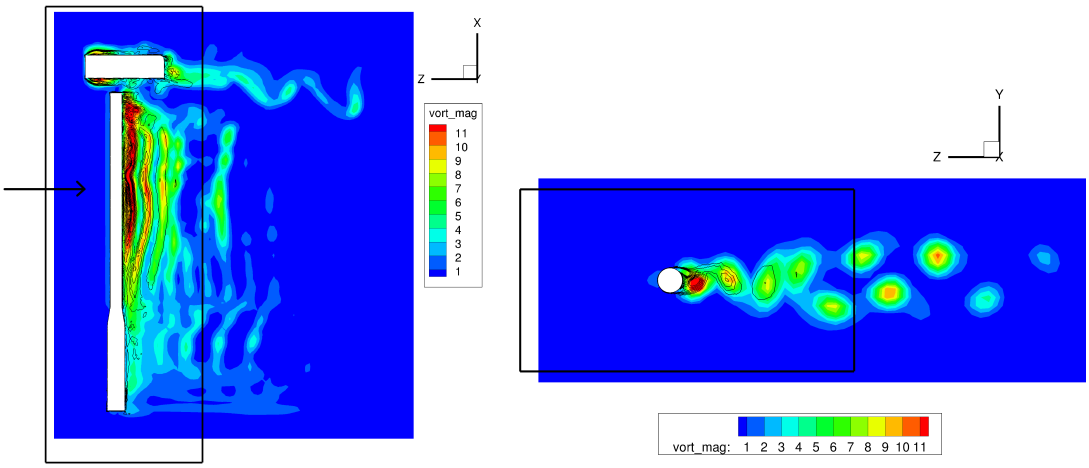
6.3.3 Tower & Nacelle

Stationary wind turbine components have been studied with the hybrid FUN3D/VorTran-M approach to verify that bluff body flow physics are accurately resolved. Instantaneous flow field vorticity magnitude is depicted in Fig. 103, with the preservation of

Table 36: Baseline wind turbine computational mixed-element mesh sizes, from Ref. 7

Component	Nodes [millions]	Cells [millions]
Tower & Nacelle	0.971	2.29
Single Blade	4.51	17.7
Background	4.78	28.3

the starting vortex observed. Since the yaw drive geometries (including the gearhead, bearing, and brake) have not been modeled, a gap between the nacelle and tower exists. This gap allows airflow to pass through and causes the nacelle starting vortex to convect faster downstream than the shed vorticity from the tower (Fig. 103a).



(a) Side view (b) Top view

Figure 103: FUN3D/VorTran-M simulated shed vorticity from the NREL6 tower/nacelle seen from the top and side, with the top viewpoint indicated by the arrow. Contour lines represent vorticity in the FUN3D domain, and the black rectangle outlines the fixed CFD domain.

Corresponding unsteady loadings for both the standalone and hybrid methods are plotted in Fig. 104. The 180° phase shift is a result of the relatively coarse wake mesh (which matches the cell size in the rotor and full wind turbine simulations discussed in subsequent sections). Since the default solution transfer scheme only inserts vorticity where the velocity field has been defined, vorticity tends to be inserted away from solid bodies, and the shed vorticity is aggregated away from the cylinder. Since the bound

vortex is unaccounted for in the wake solution (Fig. 105), the induced velocities and consequently the loading direction is reversed. This may be mitigated in the future by refining the wake mesh and/or allowing vorticity insertion at cells enclosing solid boundaries.

While the preservation of wake effects achieved by VorTran-M increases the magnitude and shifts the phase of the side forces (C_y) on the tower, both simulations capture realistic physics with vortex shedding at a Strouhal number (0.29 and 0.31 for FUN3D and FUN3D/VorTran-M, respectively) that is within the range of experimentally determined Strouhal number data in the turbulent transition range for a simulated Reynolds number of 10^6 [150]. Despite variations in amplitude, frequency content, and the 180° phase shift in the loading, the resulting cylinder drag coefficient is 0.59, which again compares well with experimental values for smooth cylinders at the given Reynolds number [151], where values range from 0.50 for a smooth cylinder to 0.63 for a nearly-smooth cylinder. While the unsteady forces are not necessarily identical between the methods, realistic physics are still captured in the simulation.

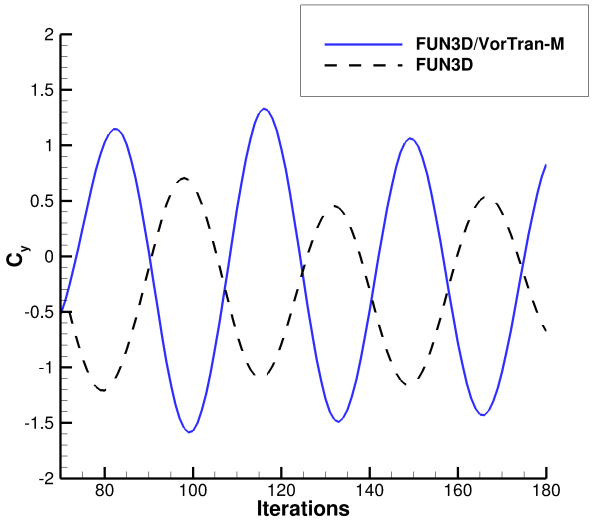
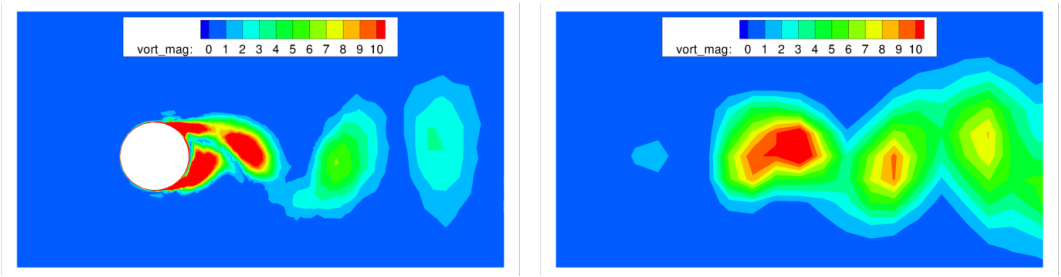


Figure 104: Comparison of hybrid and standalone CFD methods on predicted NREL6 tower side loads.

An additional important characteristic of the flow is the location of separation



(a) FUN3D solution (b) VorTran-M solution

Figure 105: A comparison of corresponding FUN3D and VorTran-M vorticity fields for the hybrid analysis at the same instant in time, on a cross-sectional slice through the tower at 4 m below hub height.

	URANS	Hybrid URANS/V-V	Experiment
Strouhal Number	0.29	0.31	0.3 ± 0.1
Drag Coefficient	0.63	0.59	0.5-0.63
Separation Location	$115 \pm 5°$	$115 \pm 5°$	120°

Table 37: FUN3D/VorTran-M predicted flow characteristics for the NREL VI Tower at $Re \sim 10^6$.

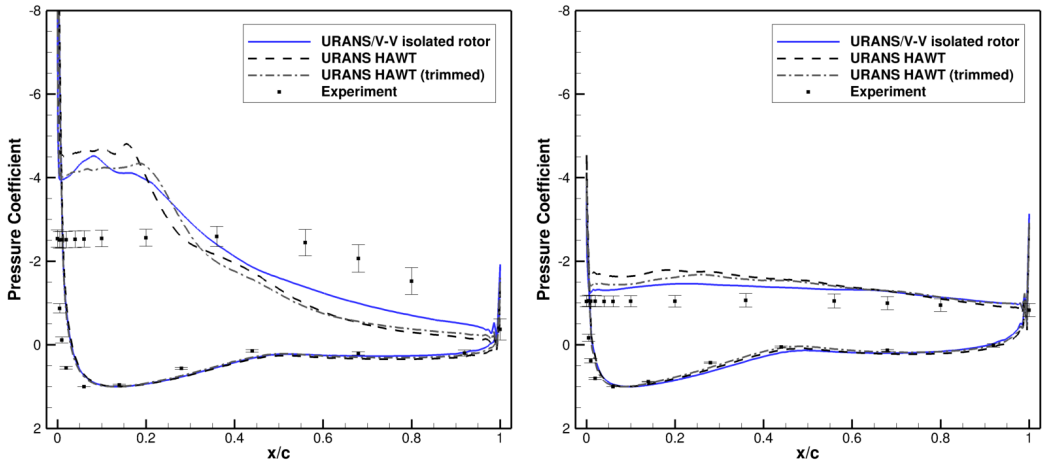
along the cylindrical tower. The location at which the skin friction changes sign indicating flow separation in the URANS simulations are approximately 115° over most of the tower length, differing from the experimental separation location by less than 5% [152]. At the top of the tower, some spatial variation and asymmetry in the separation line location is observed, resulting from interactions with vorticity shed from the nacelle. All computationally and experimentally determined flow statistics are summarized in Table 37.

6.3.4 Baseline Isolated Rotor

The next part of the hybrid analysis considers bodies in a rotating frame. Rotors in axial flow were studied, indicative of a wind turbine operating at 0° yaw or a helicopter in vertical flight. Since both FUN3D and VorTran-M deal with velocities in an inertial frame of reference, minimal modification to the interface was necessary to

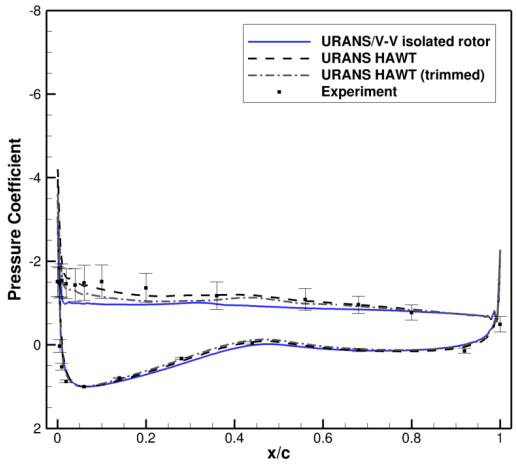
enable hybrid rotor simulations. For maximum efficiency, all cases studied were for a single-bladed axisymmetric rotor configuration (Fig. 4c). At each time step, the blade solution is duplicated and rotated by an azimuthal angle of 180° to model a two-bladed wake as in the Cardonna Tung validation case (Section 6.2.2). The baseline wake mesh spacing has been set to a nominal value of 0.14 (with reference length of one meter) and sensitivity to this parameter is further discussed in Section 6.4. To facilitate correlations with previous work [35, 7], the baseline upwind configuration ("Sequence S") was studied at 15 m/s wind speed using the same computational meshes (depicted in Fig. 101). A time step size of 0.5° per step was simulated with a temporal error tolerance of 5%, as recommended in Ref. 7. This tolerance value implies that at the end of a time step the estimated temporal error would be reduced to 5% (or less) of the error at the beginning of the time step. While the traditional URANS studies met this error tolerance after 12 subiterations on average, the error estimate [153] in the hybrid simulations tended to decrease more slowly. 40 subiterations were needed to reduce the mean flow residuals to 1% of their original value, and turbulence residuals to below 10%. However, the turbulence residuals were generally at least an order of magnitude less than the mean flow residuals at the beginning of the iteration, therefore the turbulence temporal error reduction is of less concern.

The numerically calculated pressure distributions at experimental radial stations ($r/R = 0.3$, 0.47, 0.63, 0.8, and 0.95) have been plotted against averaged experimental data in Fig. 106, with all computations using the experimental pitch setting of 3° unless otherwise noted. The full horizontal-axis wind turbine (HAWT) configuration included a tower, nacelle, and two rotor blades. These URANS data are a subset of the FUN3D computed results from Ref. 7, in which a manual trim procedure resulted in a final pitch setting of 4°. The effect of increasing the blade pitch is to turn the rotor blades into the wind and reduce blade loads. For this operating condition, an increase in tip pitch angle of 1° corresponded to a decrease in pressure coefficient

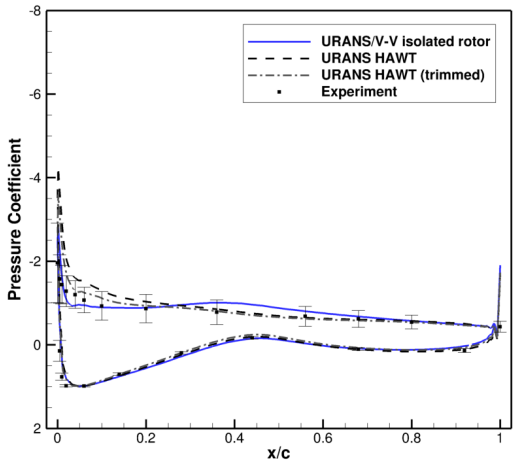


(a) $r/R = 0.30$

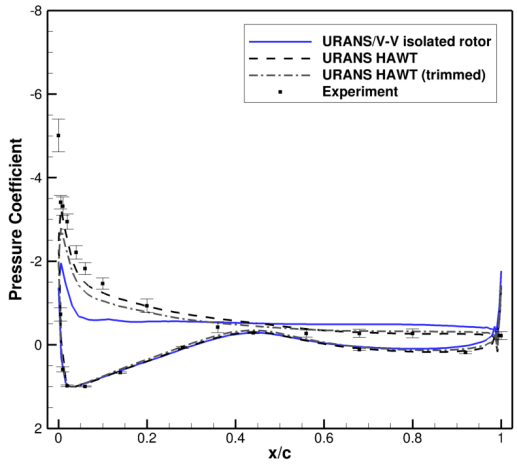
(b) $r/R = 0.47$



(c) $r/R = 0.63$



(d) $r/R = 0.80$



(e) $r/R = 0.95$

Figure 106: Comparison of pressure coefficients at radial stations corresponding to experimental measurements. The hybrid simulations only models an isolated rotor while the standalone CFD models the rotor, nacelle, and tower.

near the leading edge of 10–15% over the outboard span of the blade (Figs. 106b–e). Otherwise, the pressure distributions remained qualitatively similar.

The hybrid URANS/vorticity-velocity results calculated by FUN3D/VorTran-M (labeled "URANS/V-V") for an isolated rotor predict similar trends as the full HAWT configuration in a standalone URANS simulation ("URANS HAWT"). In general, predictions for both traditional and hybrid URANS approaches improve both qualitatively and quantitatively at stations mid-blade ($r/R = 0.47$) and outward. Hybrid results tend to predict a less-distributed suction peak and flatter pressure distributions indicative of separated flow. At $r/R = 0.8$, the hybrid calculations on the suction surface are entirely within the bounds of the experimental data (Fig. 106). However, near to the blade tip, the leading-edge suction is consistently under-predicted in comparison to the standalone URANS approach. In the following subsections, the alternate configurations are considered.

6.3.5 Overset Rotor

A hybrid-overset configuration is more general than the axisymmetric setup and allows for simulation of more varied operating conditions including yawed flow and atmospheric turbulence. An isotropic cylindrical container grid with spacings comparable to the wake mesh ($\Delta s \approx 0.14$) was generated to encompass the swept area of the rotor. This intermediate background grid contained only 84k nodes which is negligible compared to the size of the near-body blade grids. Data is inserted from the stationary FUN3D grid into the VorTran-M wake as near-blade vorticity from two blades, and data is transferred back to FUN3D as induced velocities on the entire outer surface of the container grid rather than the moving blade grids.

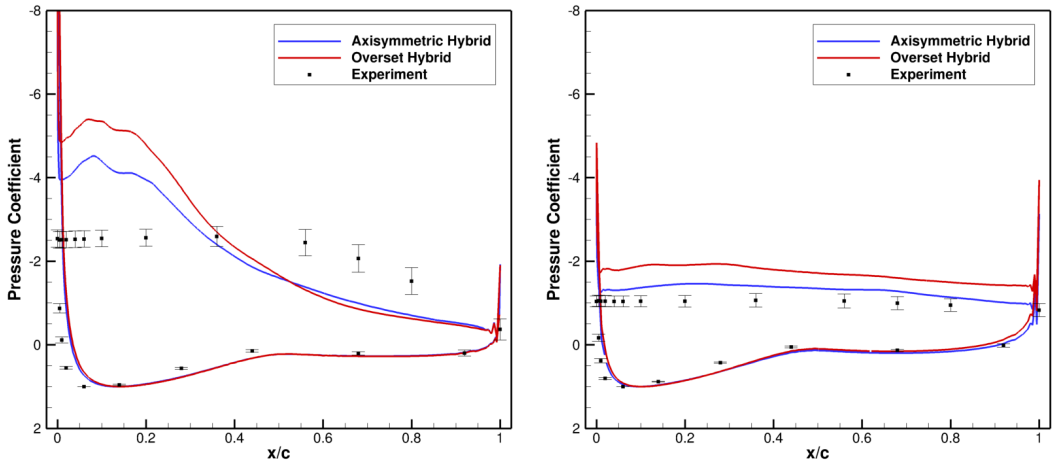
In this methodology, inadequate reduction of temporal error had a significant impact on loads. Initially, the error controller ended each time step after 7 subiterations on average when the solution had dropped by over an order of magnitude;

these results saw airload errors of nearly 40% in predicted thrust and over 250% in predicted torque. After decreasing the error tolerance to 1%, the solver performed 32 subiteration steps on average and the loads were comparable to the other approaches discussed in this section. Further discussion of integrated airloads will be presented in Section 6.5.3.

Trends in pressure distribution predicted by both hybrid approaches are comparable (Fig. 107), with the overset hybrid approach consistently over-predicting suction-side pressure compared to the axisymmetric hybrid case. At radial station $r/R = 0.63$, this shifts the pressure distribution closer to experiment but predictions worsen inboard. The solutions near the tip display similar trends, with pressure under-predicted near the leading edge and over-predicted near the trailing edge.

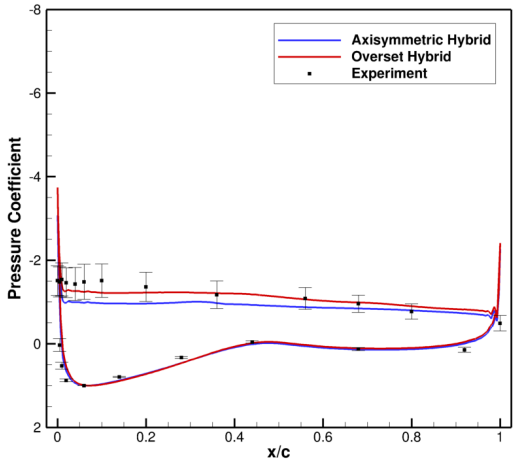
6.3.6 Refined Rotor

Time-dependent, feature-based grid adaption [110] has been performed on the original rotor blade grid. The adaption metric used was an average of vorticity scaled by edge length; time-dependency was included using a vorticity magnitude Hessian [145]. The adaption tolerance was set as the smallest adaption metric from regions where grid refinement might be desired to improve resolution of flow phenomena. The metric was calculated by sampling the flow field predicted by the original grid in the near-wake near the radial locations at which C_p data were measured. Sampled values ranged from approximately 0.3 to 0.75 (normalized by rotor tip velocity and a reference length of 1 m). The locations at which the metric were lowest to highest are the blade mid-span ($\omega \times \ell$=0.3), blade tip (0.375), blade root (0.44), and finally the 63% station (0.75). The lowest tolerance of 0.3 (corresponding to the blade mid-span at $r/R = 0.47$) was chosen, and the resulting mesh size increased by 30% from 4.51M to 5.86M nodes. Solutions on the refined grid were obtained by starting the calculation from scratch.

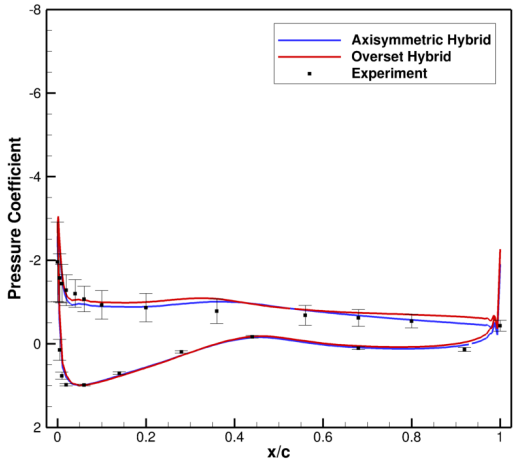


(a) $r/R = 0.30$

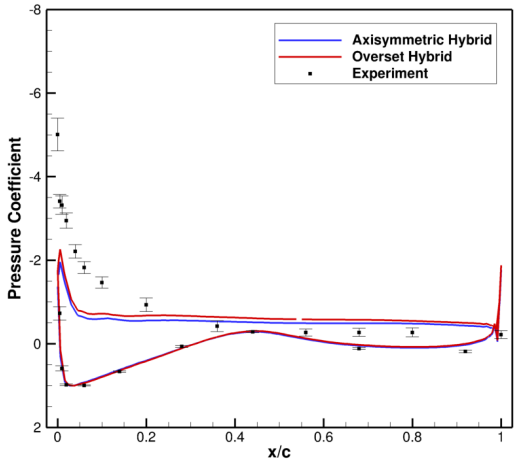
(b) $r/R = 0.47$



(c) $r/R = 0.63$

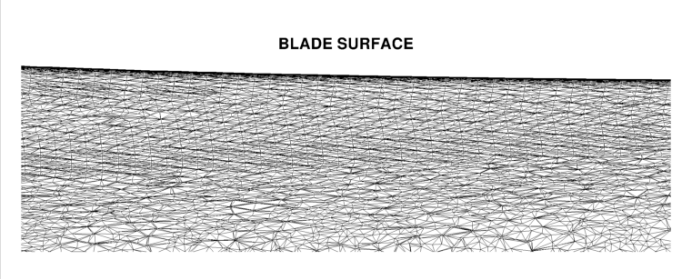


(d) $r/R = 0.80$

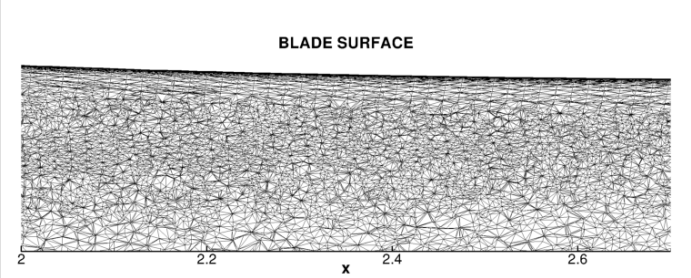


(e) $r/R = 0.95$

Figure 107: Comparison of pressure coefficients for an isolated rotor simulated with the axisymmetric and overset hybrid methodologies.



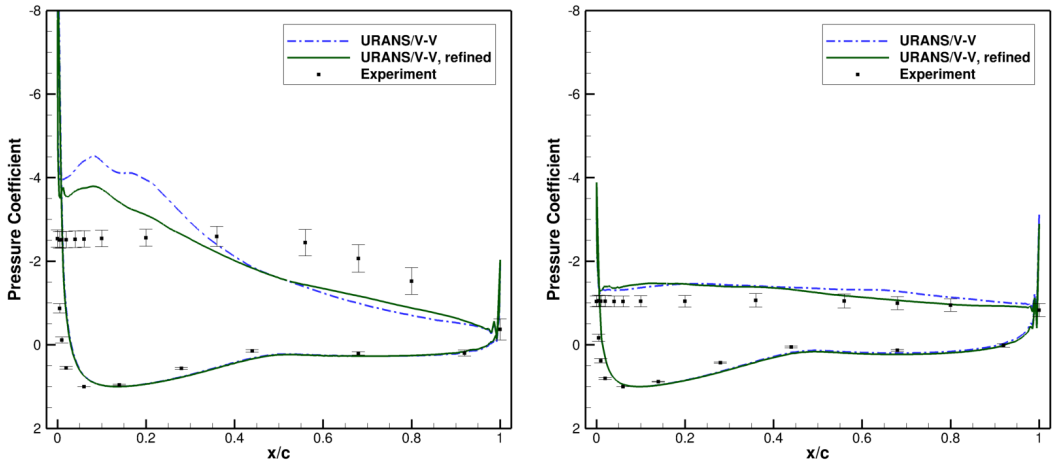
(a) Before Adaption (4.51M nodes)



(b) After Adaption (5.86M nodes)

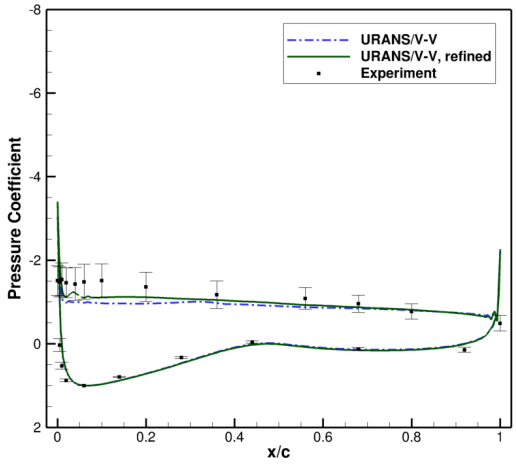
Figure 108: Slice through the NREL6 computational meshes seen above the rotor, before and after vorticity-based grid adaption.

Figure 108 compares the sectional pressure coefficients before and after grid adaption. The improved grid contained no evidence of focused refinement or redistribution in any particular area, suggesting that the baseline 4.5M node mesh already had a reasonably refined wake region. The most apparent effect of the adaption aside from point insertion was that the adapted cells became more isotropic, with striated patterns seen on the original grid (Fig. 108a) but not on the adapted grid (Fig. 108b). While the most refinement is expected in the regions where the metric is above the specified tolerance, the increased wake resolution in those regions does not necessarily guarantee improved sectional loads at those radial stations. The overall observed effect on section pressure distribution was increased suction closer to the leading edge and decreased suction closer to the trailing edge. This correction to pressure distribution changes sign at a location between 30% and 40% of the chord (e.g., in Figs. 109b, d, and e). The exceptions to this trend are at the station furthest inboard (Fig. 109a) where the pressure correction acts in the opposite direction to improve the pressure

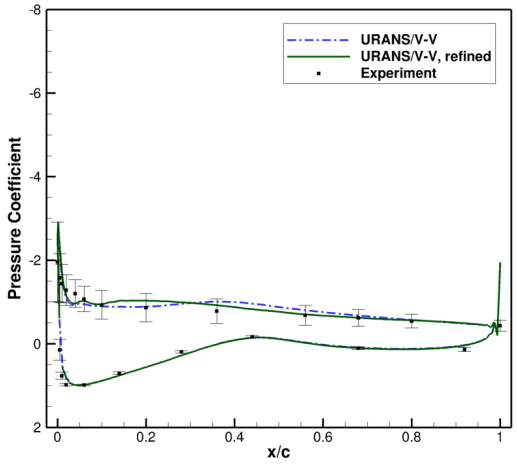


(a) $r/R = 0.30$

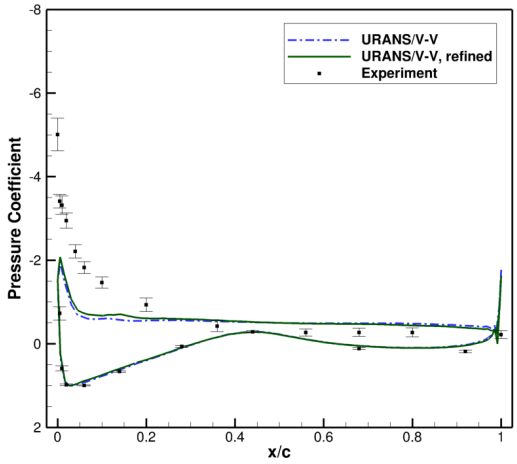
(b) $r/R = 0.47$



(c) $r/R = 0.63$



(d) $r/R = 0.80$



(e) $r/R = 0.95$

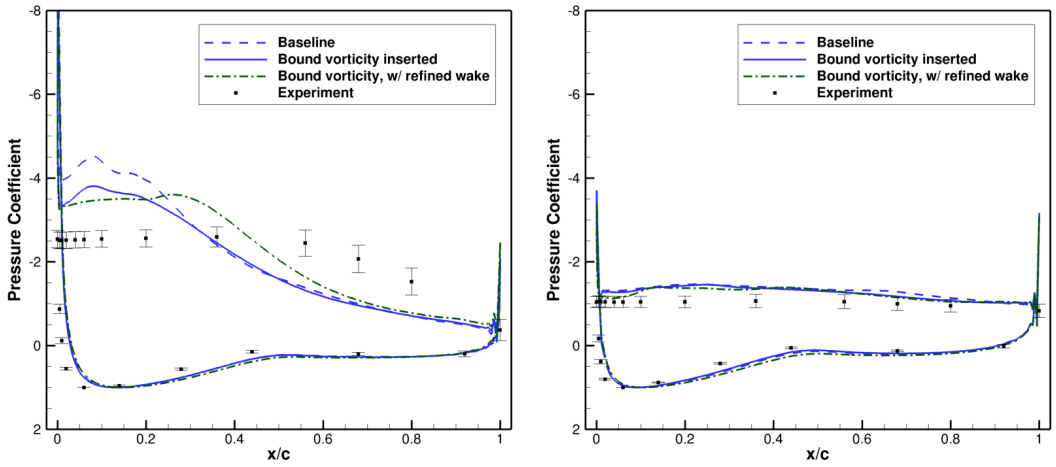
Figure 109: Comparison of pressure coefficients for an isolated rotor at radial stations corresponding to experimental measurements using baseline and refined grids.

distribution toward experimental values, and at $r/R = 0.63$ where suction is increased toward experimental values over the entire chord. Apart from the root section, the most significant pressure improvement was observed at $r/R = 0.47$ (the region on which the adaption metric is based) and $x/c \approx 0.7$, where the error between hybrid calculations and experiment was reduced from 27% to 6%. Improvements were also observed at $r/R = 0.8$ and $x/c \approx 0.5$, where error was reduced from 24% to 7%. More modest improvements were observed at $r/R = 0.63$ and $x/c \approx 0.2$, with error reduced from 29% to 18%.

6.4 Isolated Rotor with Bound Vorticity

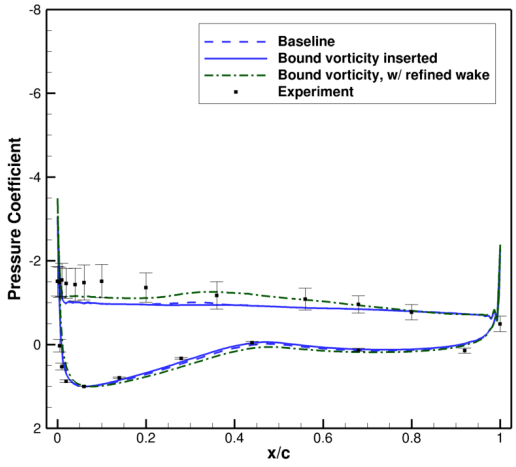
To further improve the accuracy of the coupling interface, select configurations were simulated with vorticity at or near the blade surface inserted into the wake, thus accounting for the effect of bound vorticity. Vorticity from cells near the blade surface may be calculated with finite differences between nodes that are both on the exterior and interior of the solid boundary. Interior nodes will have zero velocity and provide the largest velocity gradients. The most notable effect of the bound vorticity on the pressure distribution was observed near the root at $r/R = 0.3$ (Fig. 110a). These rather subtle corrections to the pressure distribution have a more pronounced effect on the sectional airloads (Fig. 111). The most significant change is to the normal force coefficient, which has root-mean-squared error reduced by 29%. This translates into improved predictions of integrated out-of-plane forces and moments (presented in Section 6.5.3). In comparison with the original rotor simulation performed with the baseline rotor blade grid, error in thrust and root flap bending are reduced by 13% and 9% respectively.

Two different levels of VorTran-M wake refinement were tested, $\Delta s = 0.14$ and $\Delta s = 0.07$. The coarse wake spacing is considered the baseline case, corresponding to approximately 70 cells per rotor diameter, 5 cells per root chord, and 2 cells per

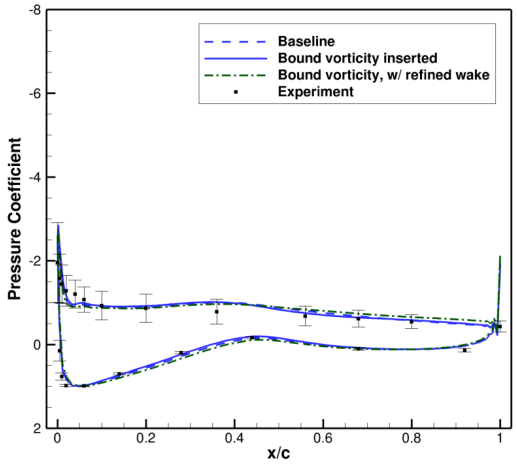


(a) $r/R = 0.30$

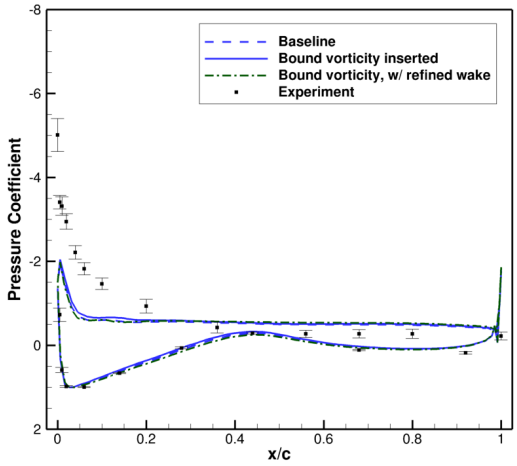
(b) $r/R = 0.47$



(c) $r/R = 0.63$

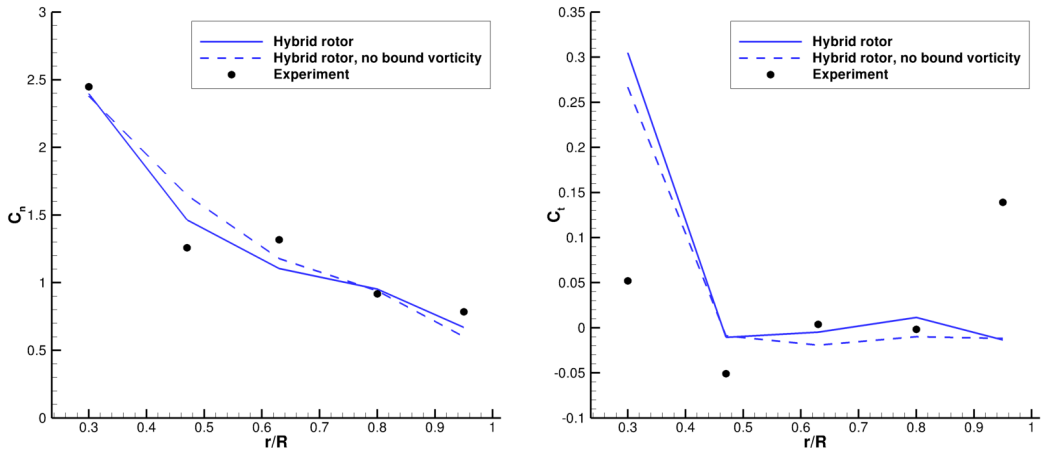


(d) $r/R = 0.80$



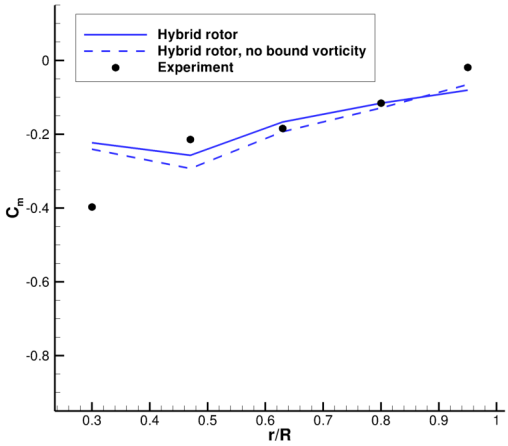
(e) $r/R = 0.95$

Figure 110: Comparison of pressure coefficients for an isolated rotor from hybrid calculations with and without bound vorticity insertion.



(a) Normal force coefficient

(b) Tangent force coefficient



(c) Pitching moment coefficient

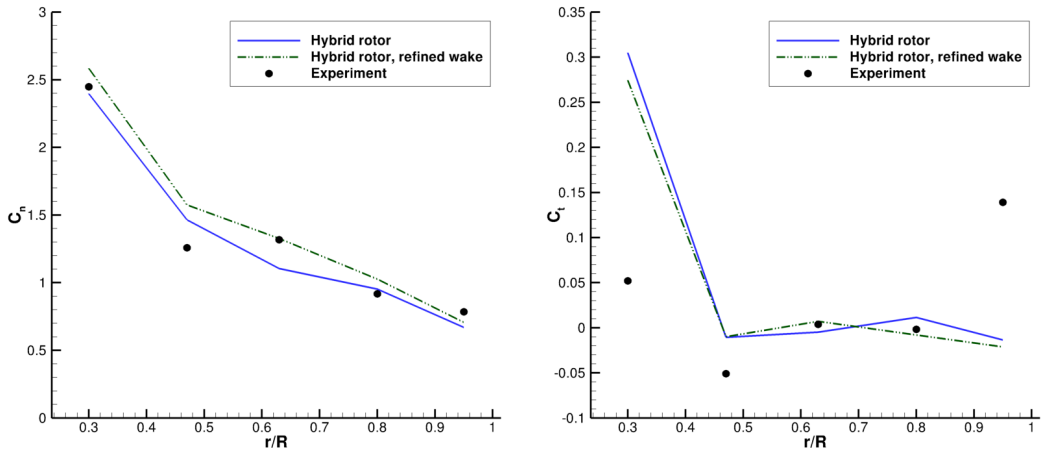
Figure 111: Sectional airloads for an axisymmetric rotor simulated with and without insertion of bound vorticity.

tip chord. This initial spacing was originally estimated from an average of near-body cell volumes. Wake resolution is comparable to recent large-eddy simulations performed by NREL and Siemens researchers who employed 64 cells across a single rotor disk in uniform flow [154]. The refined case doubled the number of cells in each direction, resulting in an overall increase in problem size by a factor of 8. As a result, the VorTran-M computation cost scaled with mesh size, although the problem size remains an order of magnitude smaller than the near-body domain.

All sectional loads are qualitatively similar between the baseline and refined wake configurations (Fig. 112). The increased wake refinement consistently increased the magnitude of the out-of-plane sectional forces and moments, i.e., the normal force and pitching moment. Improvements to sectional normal force at $r/R = 0.6$ correspond to the improved pressure distribution (illustrated in Fig. 110c), while inboard variations in pressure distribution had a minimal effect on the sectional loads due to the lower resultant wind velocity near the root. Tangent force results remain consistent with other FUN3D predictions obtained in the present study with over-predictions near the root and under-predictions at the tip, Due to the similarity in results and added cost, the refined wake spacing has not been further evaluated.

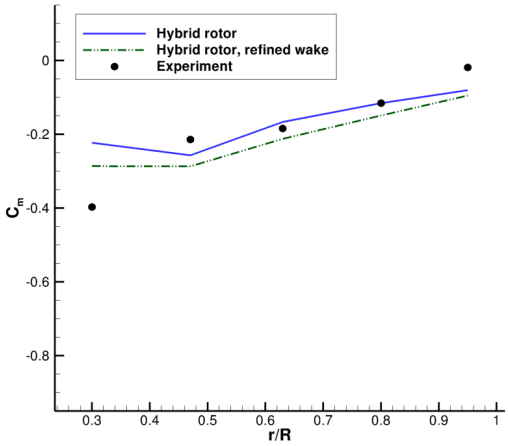
6.5 Full Wind Turbine Configuration

The hybrid methodology has been applied to completely decouple all the near-blade flow-through boundaries in the URANS domain (two blade grids and the tower/nacelle grid). To facilitate comparisons with the other test cases presented in this section, the full turbine was modeled with both baseline and refined grids with the default vorticity insertion scheme, as well as with baseline grids accounting for the effects of bound vorticity. All cases applied the standard HRLES turbulence model. A composite grid (illustrated in Fig. 4d) was formed by concatenating mixed-element meshes into a single grid. Additional inputs were the number of nodes per component mesh



(a) Normal force coefficient

(b) Tangent force coefficient



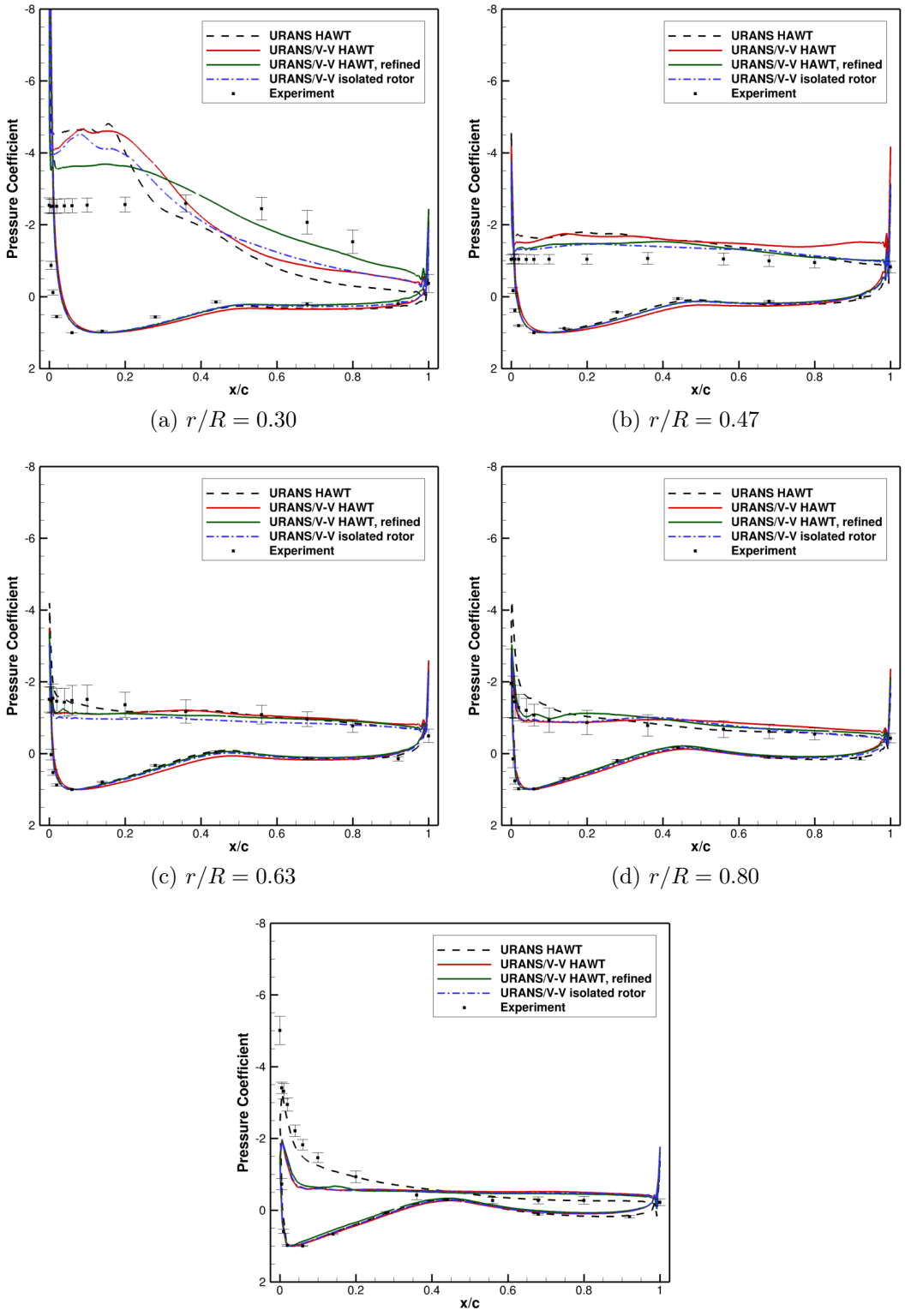
(c) Pitching moment coefficient

Figure 112: Sectional airloads for an axisymmetric rotor simulated with two different wake spacings.

and a corresponding body identification number. This body ID allows grids to be grouped together and differentiates between moving and stationary bodies, following the Suggar++ `imesh` convention wherein static meshes are denoted by `imesh=0`.

Predicted pressure coefficients for the traditional URANS and hybrid URANS/V-V full HAWT analyses are compared with the hybrid isolated rotor simulation in Fig. 113. As with the previous isolated rotor simulations, predictions near the blade root ($r/R = 0.3$) typically misrepresent the stall behavior of the blade. At stations closer to mid-span ($r/R = 0.47$ and 0.63), the baseline decoupled boundaries analysis tend toward the URANS HAWT result, and at $r/R = 0.63$ there are improved pressure predictions near the leading edge. The baseline hybrid HAWT result is more similar to the full URANS result that the isolated rotor because the nacelle modifies the geometry of the inboard rotor wake, as illustrated by Fig. 114. While the isolated rotor predicts a well-defined root vortex, the presence of the nacelle in the full wind turbine calculation pushes the root vortex outboard where it is distorted by the unsteady blade downwash. At the stations furthest outboard ($r/R = 0.80$ and 0.95), the influence of the nacelle is minimized and the hybrid solutions are virtually identical (Fig. 113e) despite differences in the instantaneous wake structure, implying that on average the unsteady wake predicted by this analysis has the same effect on integrated airloads near the tip.

When considering the full HAWT configuration with refined rotor blade grids, the outboard trends remain the same. However, notable improvements are observed on the suction side on the two most inboard stations ($r/R = 0.30$ and 0.47). Out of all simulations, this analysis comes the closest to predicting the pressure distribution at $r/R = 0.30$ in terms of both the general trend and the pressure magnitude on the trailing half of the blade where the flow appears attached. At $r/R = 0.47$, the refined hybrid HAWT result starts to approach the experimental prediction rather



(a) $r/R = 0.30$

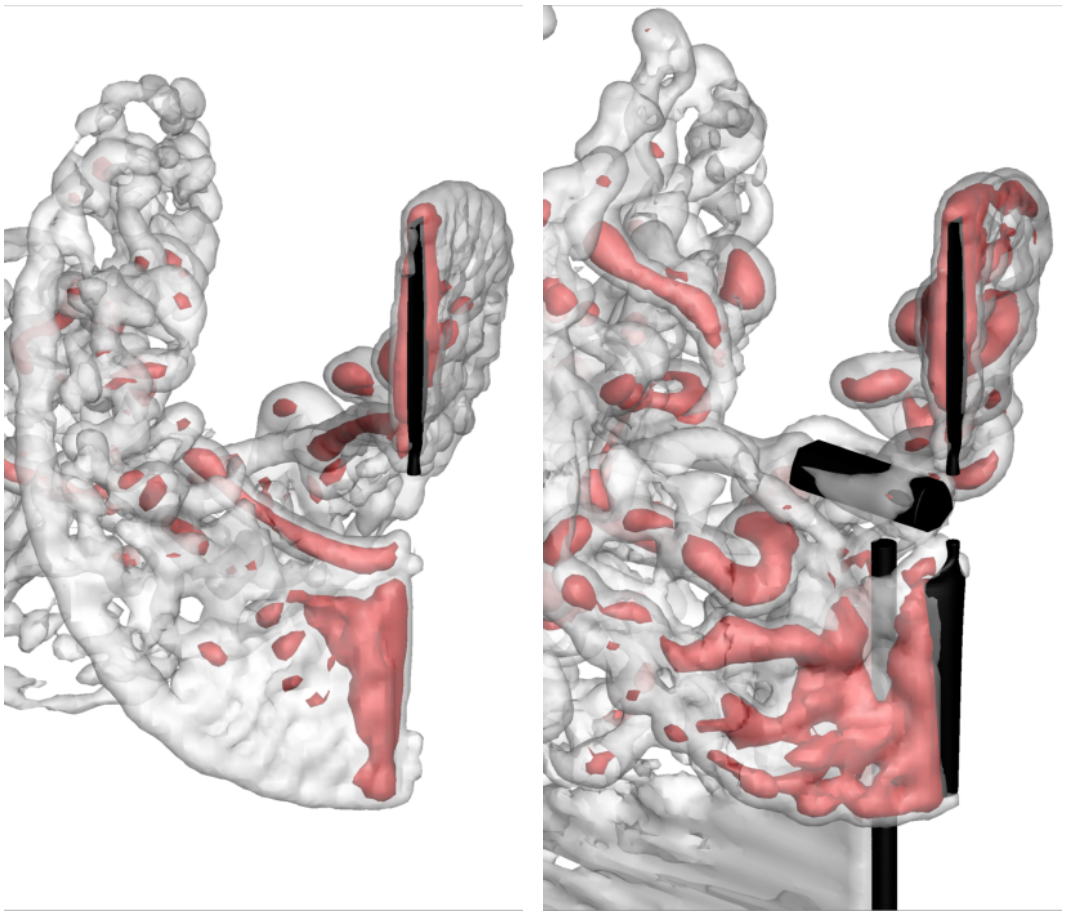
(b) $r/R = 0.47$

(c) $r/R = 0.63$

(d) $r/R = 0.80$

(e) $r/R = 0.95$

Figure 113: Comparison of pressure coefficients at radial stations corresponding to experimental measurements for an isolated rotor and full turbine configurations using traditional URANS and hybrid decoupled boundaries approaches (without bound vorticity insertion).



(a) Isolated Axisymmetric Rotor (b) Full Wind Turbine

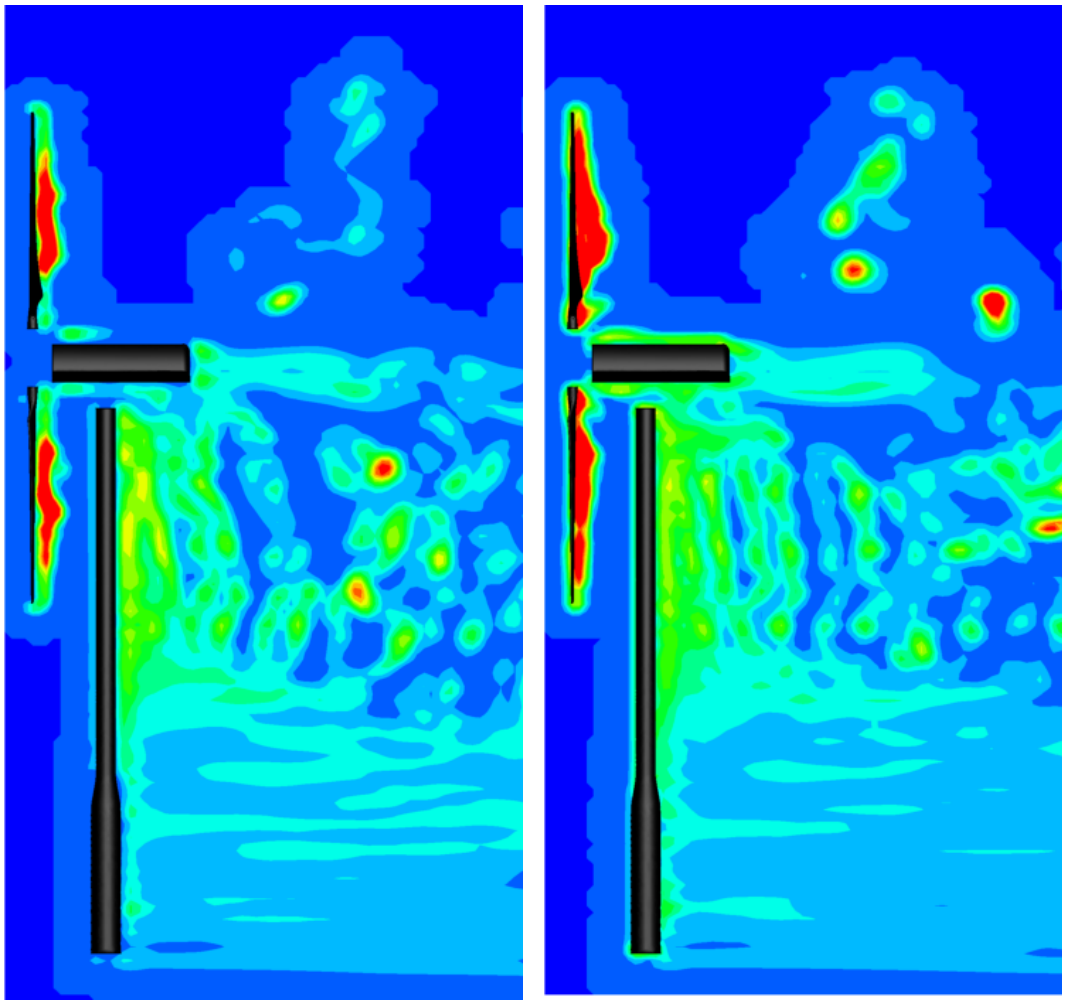
Figure 114: Two iso-contours of vorticity magnitude for hybrid analyses of the NREL Phase VI turbine; the isolated rotor assumes axisymmetry while the full turbine applies the decoupled boundaries technique.

than the full URANS result. Finally, at $r/R = 0.63$, rather than predicting a well-defined suction peak (as with the standalone URANS analysis), the predicted pressure distribution is within experimental bounds and closer to the experimentally observed stall behavior.

6.5.1 Full Wind Turbine with Bound Vorticity

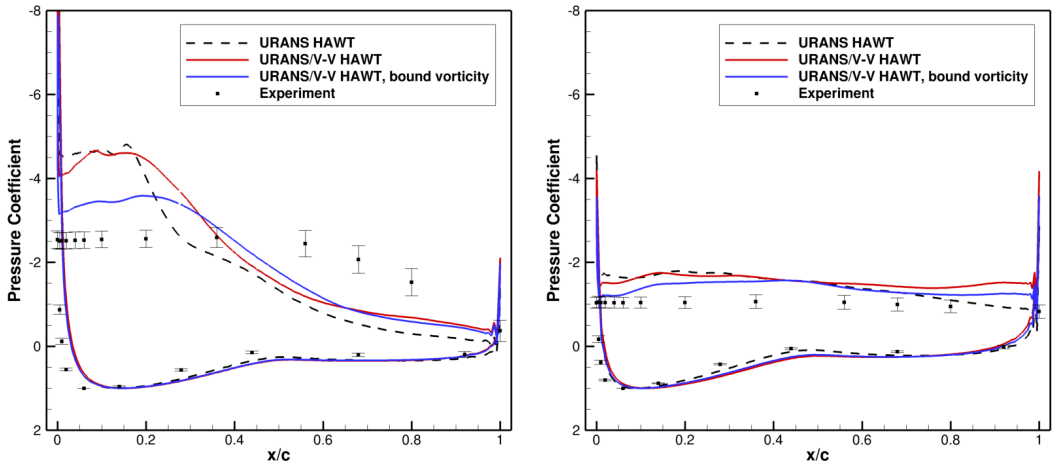
Simulations of the full turbine accounting for bound vorticity were also performed. The effect of the additional inserted vorticity is illustrated in Fig. 115, wherein the high magnitude near-blade vorticity appears over the entire span of the blade. In addition, the nacelle also has increased levels of vorticity around its leading edges. These differences in the flow field improve pressure predictions inboard (Fig. 116a and b) but have a minimal effect on the section (Fig. 117) and integrated loads (presented in Section 6.5.3). In the results presented in Figs. 116 and 117, "URANS HAWT" (or "URANS, full config") refers to the reference overset FUN3D analysis performed by Lynch and Smith [7]. "URANS/V-V HAWT" (or "Hybrid, full config") refers to the current hybrid simulations with the decoupled boundaries technique. "Bound vorticity" indicates that the hybrid interface accounts for near-body vorticity and transfers solutions to the wake solver at locations where one or more nodes are inside of a solid surface. In the presentation of sectional loads (Fig. 117), both "Hybrid" data sets include the effects of bound vorticity.

Sectional loads may be compared to elucidate differences between the hybrid and standalone URANS simulations, as well as differences between simulating an isolated rotor and a full wind turbine configuration (Fig. 117). Good agreement is observed between the hybrid and standalone URANS calculations of the full turbine in terms of normal force coefficient. Errors were less than 10% except at mid-span ($r/R = 0.47$). In terms of tangent force, both hybrid and standalone URANS methodologies are unable to capture the root and tip behaviors, with standalone URANS predicting similar

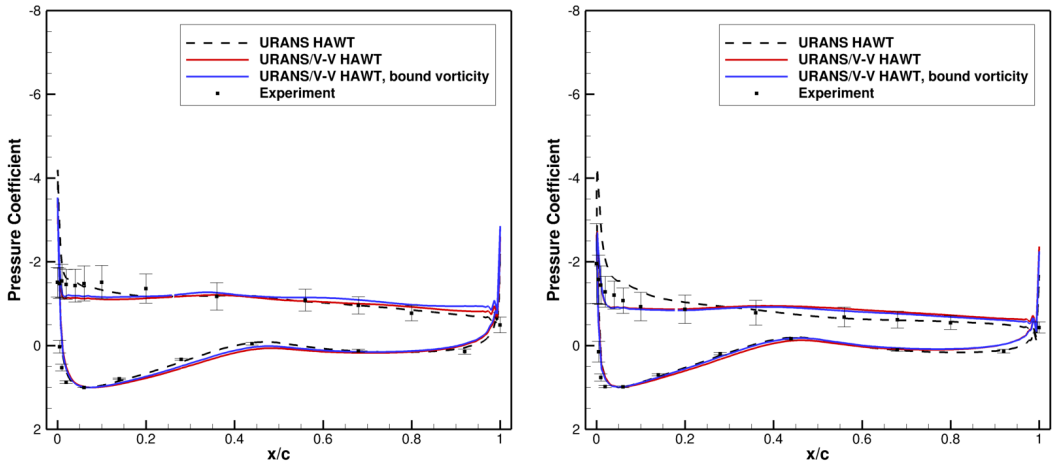


(a) Baseline (b) With bound vorticity

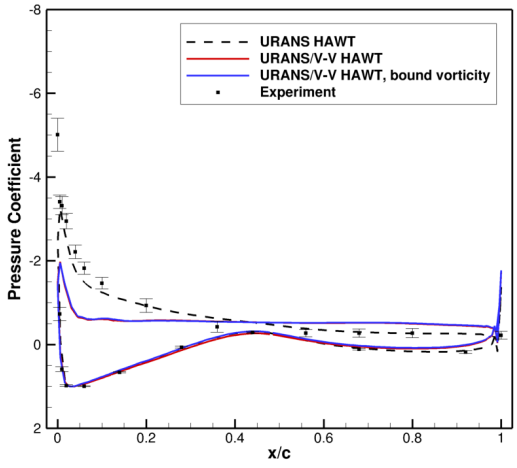
Figure 115: Contours of vorticity magnitude for the NREL Phase VI turbine computed without and with vorticity insertion at solid surfaces.



(a) $r/R = 0.30$ (b) $r/R = 0.47$



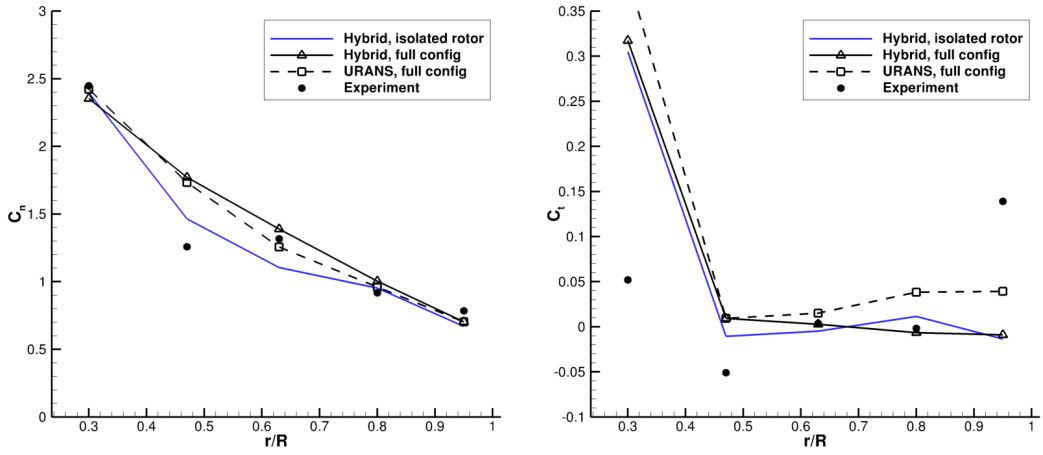
(c) $r/R = 0.63$ (d) $r/R = 0.80$



(e) $r/R = 0.95$

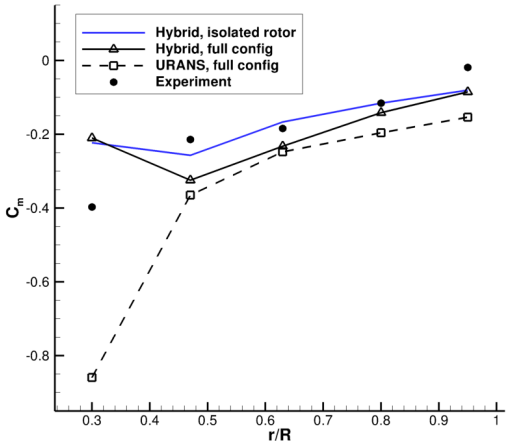
Figure 116: Comparison of pressure coefficients at radial stations corresponding to experimental measurements for a hybrid full turbine configuration with and without the insertion of near-body vorticity.

trends with higher magnitude forces than the hybrid approaches. The magnitude of the pitching moments are also overpredicted by the standalone URANS methodology. For both the tangent force and pitching moment coefficients, the hybrid solutions are similar near the root and at the tip. However, the hybrid full turbine forces tend toward the URANS result at mid-span and inboard, indicating the influence of the tower and nacelle geometry.



(a) Normal force coefficient

(b) Tangent force coefficient



(c) Pitching moment coefficient

Figure 117: Sectional airloads for a hybrid full wind turbine configuration (with the insertion of near-body vorticity) in comparison with a hybrid isolated rotor.

6.5.2 Comparison with Other Methodologies

An analysis has been performed on four different quantities highlighted by NREL in a blind study [155]: normal force, tangent force, pitching moment, and local dynamic pressure. Since the present hybrid results with the FUN3D/VorTran-M methodology were computed without any *a posteriori* knowledge (e.g., no trim procedure was applied), the results may be readily compared with both experiment and other Navier-Stokes CFD results from the blind study. Comparison results were generated by other researchers using EllipSys3D, ADDWANS, and GT-Hybrid. EllipSys3D [156] solves the axisymmetric, incompressible unsteady Reynolds-averaged Navier-Stokes (URANS) equations using second-order dual time-stepping and second-order upwind flux reconstruction. The rotor is modeled as an actuator disc with a nonlinear filtering algorithm applied to prevent spurious oscillations at the rotor disc. ADDWANS uses the same Navier-Stokes methodology but applies blade element momentum theory to add forces to the flow field [157]. Blade tip effects are modeled using Prandtl's tip-loss correction. Lastly, GT-Hybrid is a blade-resolving compressible URANS methodology coupled to free-wake model in the far field. The URANS solver is based on a third-order upwind reconstruction scheme and a second-order time-stepping scheme.

All comparison results have been averaged over a partial revolution of airloads data. Stand-alone rotor results computed by the hybrid FUN3D/VorTran-M methodology at two levels of grid refinement are compared to FUN3D and FUN3D/VorTran-M simulations of the full wind turbine, illustrating differences between the traditional URANS methodology and the hybrid URANS/vorticity-velocity (V-V) approach, as well as to demonstrate potential improvements due to grid refinement. The distribution and average of the results obtained by other researchers have been plotted as symbols with error bars.

Normal Force Coefficient

Excellent correlations were obtained by the FUN3D analyses in terms of normal force coefficient C_n (Fig. 118), defined as the force normal to the airfoil section in the downstream direction. Despite the large scatter in observations reported by other researchers, most notably at the root section, all of the FUN3D calculations are within 10% of the experimental results except at $r/R = 0.47$. Grid refinement consistently improves the hybrid result, reducing the maximum error from 31% to 20%. While none of the calculations were able to reproduce the inflection point between the $0.47R$ and $0.63R$ stations, all FUN3D solutions are within 11% of experiment at $0.63R$, outperforming all other CFD codes. Since the suction-side pressure peak is generally underpredicted at outboard stations by the hybrid isolated rotor analyses (Fig. 106), the normal force at $r/R = 0.95$ is also slightly underpredicted compared to the traditional URANS result. The baseline hybrid full turbine calculation matches the trend of the standalone full FUN3D and hybrid rotor calculations, but generally overpredicts C_n. Grid refinement for the hybrid full configuration improved results overall, with the most significant error reduction of 26% at $r/R = 0.47$. The most notable improvements due to grid refinement were observed at this location, where it is closest to the experimental value. This is the region on which the grid adaption metric was based (described in Section 6.3.6). In addition, the error at $r/R = 0.95$ is reduced from 23% to 17%, and the error at $r/R = 0.3$ is reduced to zero.

Tangent Force Coefficient

The tangent force coefficient is defined as the force aligned with the airfoil section in the direction of rotation. While predictions at $r/R = 0.63$ are within 20% of the maximum experimentally observed tangent force (C_t), correlations are generally poor among all methodologies at the root and tip stations, as evidenced by the range of predicted values (Fig. 119). On average, the comparison codes tended to slightly

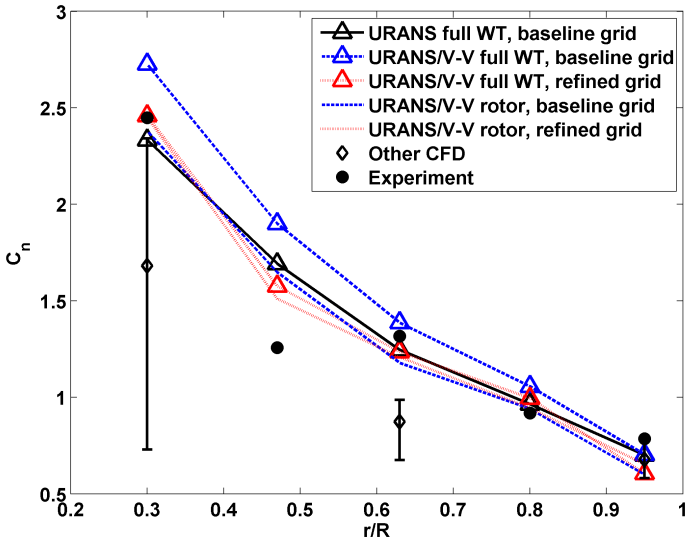


Figure 118: Averaged normal force distribution calculated by hybrid FUN3D/VorTran-M compared with results obtained from experiment and a blind study [155].

overpredict C_t at the root and underpredict C_t at the tip, and the FUN3D results also follow this trend. The hybrid methodology improves the root prediction somewhat in the isolated rotor case, although the tangent force still differs from experiment by a factor of five or greater. The best prediction resulted from the refined, hybrid full turbine analysis and agrees with the improved pressure prediction observed in Section 6.5. In this case, the tangent force constitutes a significant component of rotor torque, and the reduced tangent force near the tip is compensated for by the increased tangent force near the root resulting in excellent correlations with integrated torque. This is in agreement with NREL findings that suggest certain methodologies had offsetting errors in inboard and outboard predictions [155]. An additional observation from Fig. 119 is that modeling the tower and nacelle geometry worsens predictions on the baseline grid but improves predictions on the refined grid.

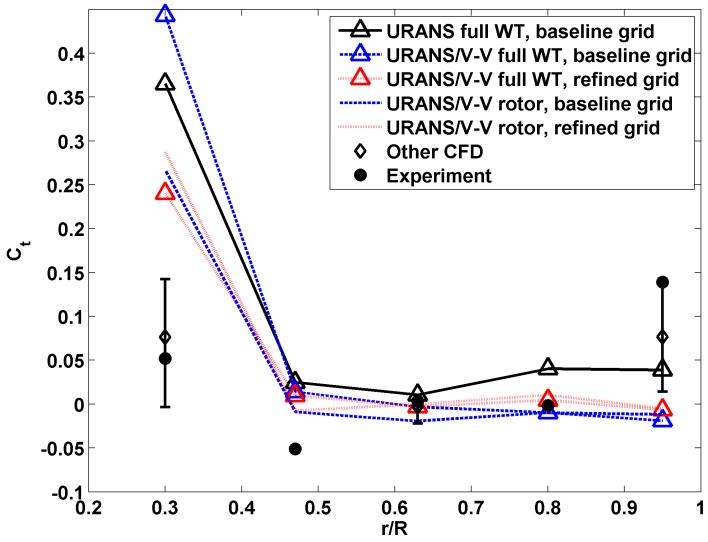


Figure 119: Averaged tangent force distribution calculated by hybrid FUN3D/VorTran-M compared with results obtained from experiment and a blind study [155].

Pitching Moment Coefficient

Out of all the computational approaches the hybrid methodology is able to closely predict the pitching moment at the outboard stations, within 3% of the maximum pitching moment amplitude at $0.63R$ and $0.8R$, and within 14% at the tip station. Predictions on the refined grid improve toward the experimental observations at the 30% and 47% stations by 5% and 11% respectively. The largest improvement again corresponded to the region on which the grid adaption metric was chosen as described in Section 6.3.6. Best overall predictions over the span of the blade (including both FUN3D and the other CFD results) were obtained by the refined hybrid full turbine analysis

The traditional URANS methodology tends to overpredict the magnitude of the pitching moment coefficient. However, it is the only methodology that is able to correctly predict the pitching moment trend near the root ($r/R = 0.3$). In comparison to the standalone URANS predictions, the hybrid calculated suction-side pressures

tend to be very flat (Fig. 106). This result alters the pitching moment by shifting the center of pressure closer to the pitch axis at $x/c = 0.3$. This suggests that the experimental values have been predicted based on different physics. The similarity between hybrid tip predictions (between the isolated rotor and full turbine as illustrated in Fig. 113) results in full-turbine pitching moments very close to the isolated rotor results and the experiment result.

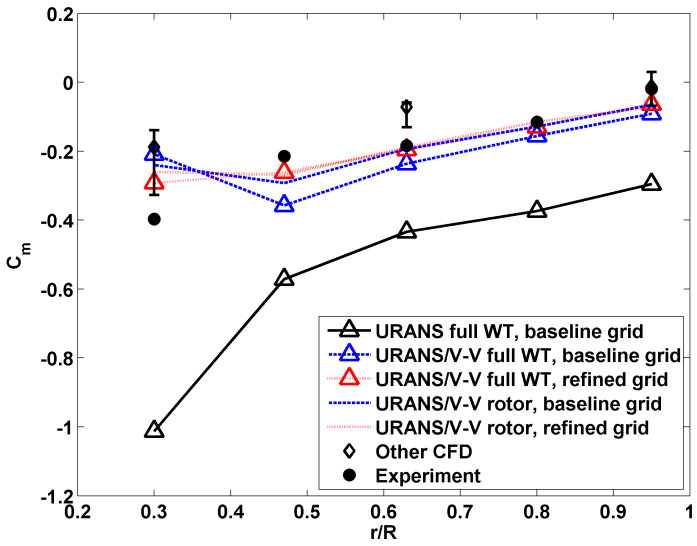


Figure 120: Averaged pitching moment distribution calculated by hybrid FUN3D/VorTran-M compared with results obtained from experiment and a blind study [155].

Local Dynamic Pressure

Lastly, the local dynamic pressure, q_{norm}, is used to normalize pressure, force, and moment coefficients. By definition, this has been chosen to be the stagnation point pressure. To replicate experimental procedure, this value has been set to the maximum pressure over the airfoil section [148]. This quantity is generally predicted to within 6% by FUN3D full turbine simulations as well as all other researchers in the blind study (Fig. 121). In comparison, the hybrid isolated rotor simulation on the baseline grid consistently overpredicts the spanwise stagnation pressure distribution

by between 14% and 25%.

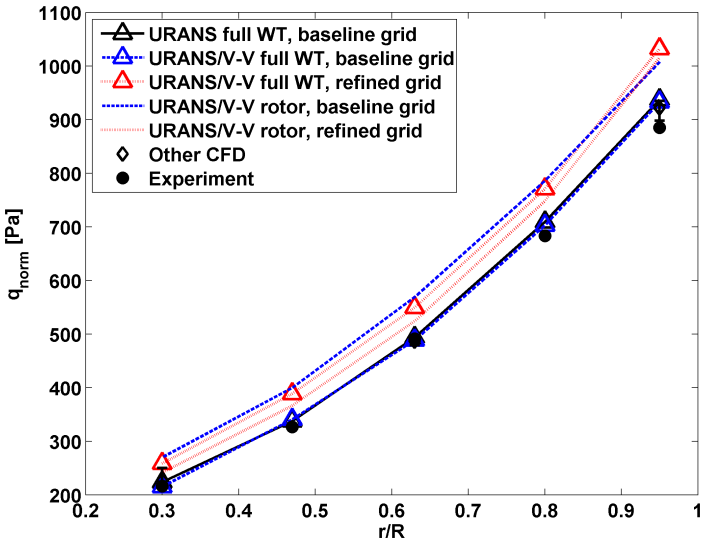
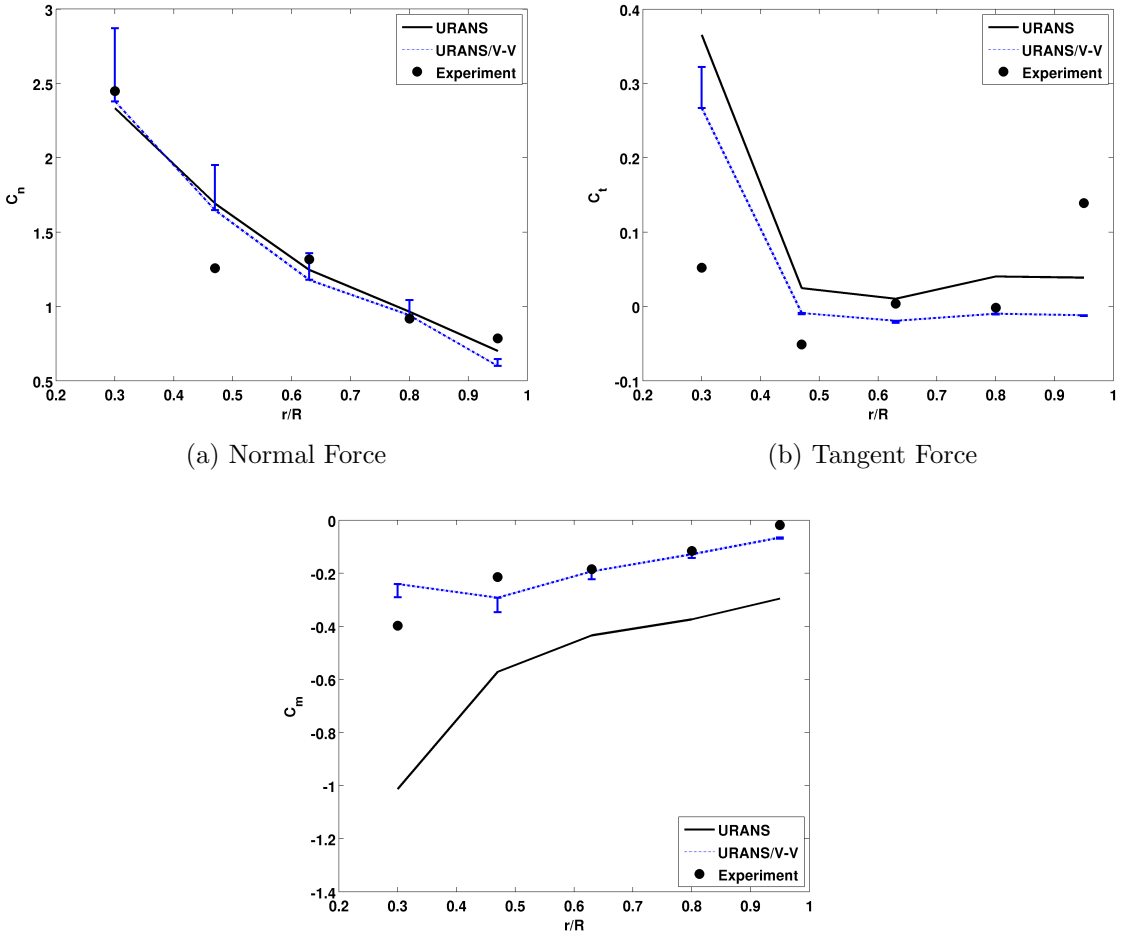


Figure 121: Averaged local dynamic pressure distribution calculated by hybrid FUN3D/VorTran-M compared with results obtained from experiment and a blind study [155]. The triangle symbols represent results from the best alternate CFD code.

Since this quantity is overpredicted, the calculated suction-side pressure coefficients normalized by this value will tend to be underpredicted, as was observed in Section 6.3.4. The sensitivity to q_{norm} of the aforementioned force and moment coefficients is illustrated in Fig. 122. Error bars indicate the resultant values from normalizing with the full URANS result which was a better approximation of experimental observations. In general, inboard stations are most sensitive to the local dynamic pressure with up to 20% variability in magnitude of sectional loads. In comparison, the tip station is only increased by 8% when using the more realistic normalization factor, which does not completely account for the observed suction peak deficiency in the hybrid isolated rotor results. Moreover, the differences between the traditional and hybrid URANS approaches can not be accounted for solely by differences in normalization since the adjusted hybrid results differ both qualitatively and quantitatively from the traditional URANS calculations.



(a) Normal Force

(b) Tangent Force

(c) Pitching Moment

Figure 122: Sensitivity of baseline hybrid isolated rotor solution (URANS/V-V) to changes in the local dynamic pressure, q_{norm}, used for normalization. For reference, the URANS solution for a full wind turbine has been shown.

6.5.3 Integrated Loads

Blade loads were integrated at pressure tap locations at five different radial locations (r/R = 0.3, 0.47, 0.63, 0.80, and 0.95) according to experimental procedures (detailed in Ref. 148). Section aerodynamic coefficients are calculated for a single blade by numerically integrating pressures in a loop around the airfoil using the trapezoidal rule. 22 integration segments are formed with unsteady pressure data from the same number of pressure taps distributed around the blade at each of the five radial locations. With the FUN3D built-in sectional force integration capability, integration loops intersected between 311 and 690 points on the blade surface depending on radial location. Locations with smaller cross-sections had fewer points in the integration loop. However, the difference between the two integration approaches was minimal. For thrust and root flap bending the resulting loads were virtually identical, while torque values were about 5% lower in general when integrating at pressure taps.

The resulting sectional pressures were resolved into forces at each radial location and then multiplied by a fraction of the blade planform. Three quantities have been calculated: thrust (T), torque (Q), and root flap bending moment (RFB). In blade-fixed coordinates (x from root to tip, y from trailing edge to leading edge, and z normal to the blade into the wind), these quantities are defined as:

$$T = 2(-C_z\, q_\infty A), \tag{91}$$

$$Q = 2(C_y\, q_\infty A r), \tag{92}$$

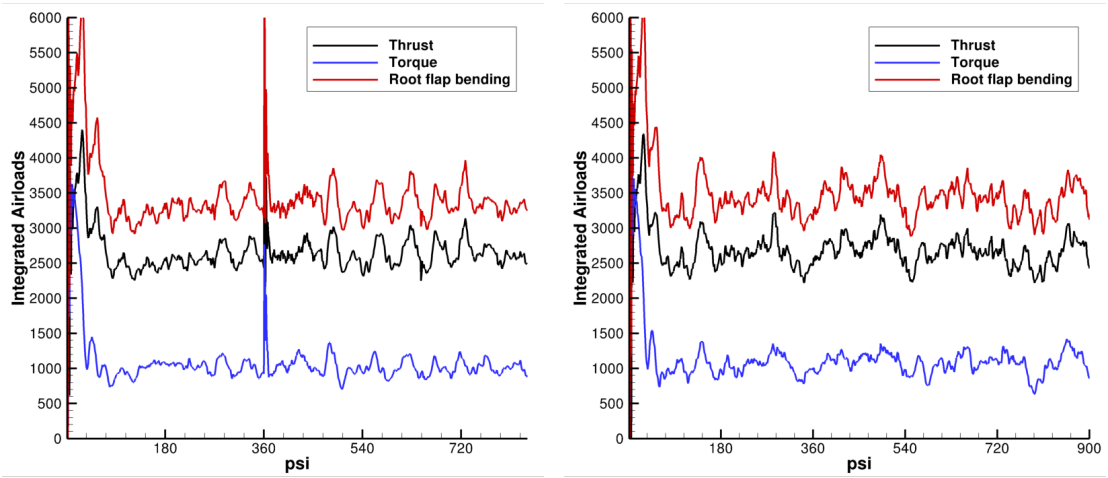
and

$$RFB = -C_z\, q_\infty A(r - 0.432) \tag{93}$$

where the radial coordinate is offset by 0.432 m, the location at which strain gauges were placed. C_x, C_y, and C_z are averaged aerodynamic force coefficients in the rotating blade reference frame (with $+x$ from root to tip, $+y$ from trailing edge to leading edge, and $+z$ normal to x, y in the upstream direction). $q_\infty = \frac{1}{2}\rho_\infty V_{tip}^2$ is the dynamic

pressure based on rotor tip speed, and A is the blade planform area associated with the blade section. Thrust and torque were multiplied by two to obtain the load contribution from both rotor blades. A summary of these three integrated rotor quantities is presented in Table 38.

Corresponding rotor airload time histories are plotted in Figs. 123 and 124 for the baseline and refined grid cases. The unsteady flow has been averaged over the final half-revolution of the simulation. Transients due to out-of-sync restart files occurred in some cases (Fig. 123a and 124b), but the solution was advanced for at least an additional half-revolution before data were used to determine the mean loads. The observed unsteady loads were highly oscillatory due to the separated nature of the flow at the 15 m/s operating condition. All simulations (including different levels of near-body and wake refinement) were run to the same level of convergence, with variations in out-of-plane forces and moments (i.e., thrust and root flap bending) of at most 8% (Table 39). Greater variations existed in the calculated torque, with a maximum standard deviation of 18% in all cases. Numerical convergence was achieved with solver residuals below 10^{-7} in all cases.



(a) Isolated rotor

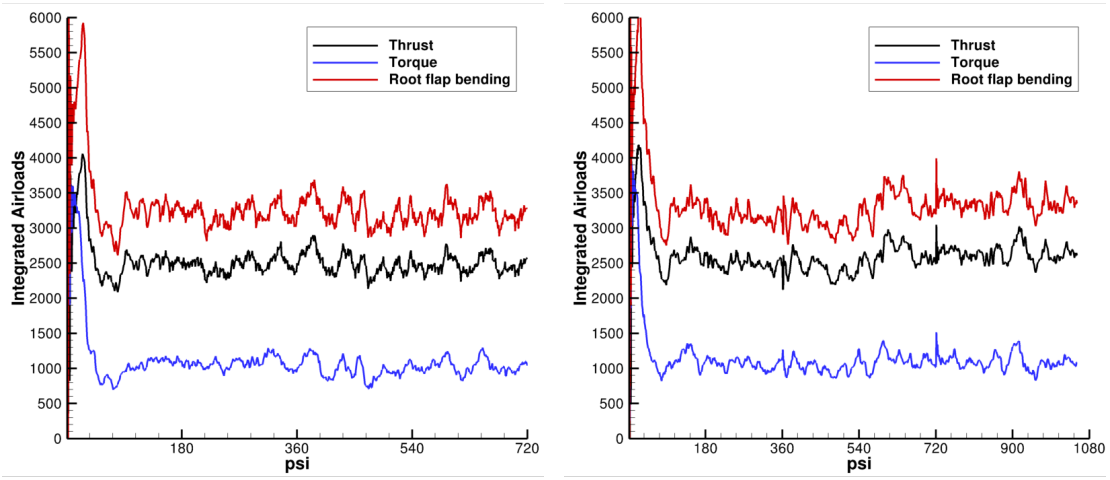
(b) Full wind turbine

Figure 123: Time history of integrated blade airloads from hybrid calculation on baseline grids.

Table 38: Mean integrated loads (thrust, torque, and root flap bending, RFB) for calculations of the NREL VI isolated rotor and full turbine (HAWT) configurations at the experimental pitch setting of 3° unless otherwise noted.

Case	V-V	Steps	Thrust [N]	Err. [%]	Torque [N*m]	Err. [%]	RFB [N*m]	Err. [%]
Experiment		–	2275	–	1104	–	3040	–
Overset rotor		900	2199	-3.47	917	-16.9	2889	-4.97
Overset rotor	x	1980	2681	17.7	1312	18.8	3460	14.0
Rotor	x	1680	2660	16.8	1017	-7.88	3361	10.6
Rotor, refined blade	x	1440	2477	8.74	1032	-6.52	3205	5.43
HAWT		3240	2409	5.75	1216	10.1	3130	2.96
HAWT, trimmed*		2520	2301	1.01	1372	24.3	2999	-1.35
HAWT	x	1800	2590	13.7	1073	-2.81	3355	10.4
HAWT, refined blade	x	2100	2645	16.1	1083	-1.90	3379	11.2

* Trimmed pitch was 4°



(a) Isolated rotor

(b) Full wind turbine

Figure 124: Time history of integrated blade airloads from hybrid calculation on refined grids.

Table 39: Maximum standard deviation of converged rotor loads over the final partial revolution out of all configurations and conditions simulated, presented as a percentage of the mean value.

	Thrust & Root Flap Bending	Torque
Isolated rotor	5.5%	9.3%
Full wind turbine	7.8%	16.8%
Isolated rotor, w/ bound vorticity	7.2%	14.5%
Full wind turbine, w/ bound vorticity	7.4%	17.6%

For the traditional URANS simulations of the full wind turbine (performed by Lynch and Smith [7]), simulating the experimental pitch setting resulted in thrust and root flap bending predictions within 6% of experiment, and torque within 10%. Trimming the rotor (increasing the blade pitch from 3° to 4°) was able to reduce the error in thrust and root flap bending, but the error in torque more than doubled. The hybrid decoupled boundaries simulation of the full turbine was able to obtain the best torque prediction (within 3% of experiment for the baseline blade grids and within 2% for the refined grids) but thrust and root flap bending errors are 8–10% higher than the corresponding traditional URANS simulation.

When considering all three aerodynamic quantities, the largest errors were observed in the hybrid overset methodology. Even after adequate reduction of the temporal error tolerance (described in Section 6.3.5), this approach had the highest errors in thrust and root flap bending, and the second highest error in torque. In this configuration, the hybrid interface is located on a stationary container grid that adds an additional level of data transfer between the near-body and container grids. In addition, the wake influence is imposed on a boundary that is further away from the blade and the effects are filtered through an overset interface which could add additional interpolation errors to the flow field. Finally, the relatively coarse container mesh is not as successful as VorTran-M in preserving flow features.

The torque predictions for the other hybrid rotor simulations on an isolated rotor were all within 12% of the experimental value (with the exception of the hybrid overset approach). This is a significant improvement over the NREL blind study in which other Navier-Stokes researchers obtained torque estimates with between 20% and 40% error [155].

Overall improvements to all three aerodynamic quantities were achieved for the isolated rotor on the refined grid. The errors in thrust and root flap bending were reduced by nearly half, and error in torque was further reduced to within 7%. However, depending on the quantity of interest, the modest improvement in torque prediction may not justify the added 30% added cost due to increased node count. For the full turbine, less than 1% improvements in torque and root flap bending are offset by a 2% increase in thrust error.

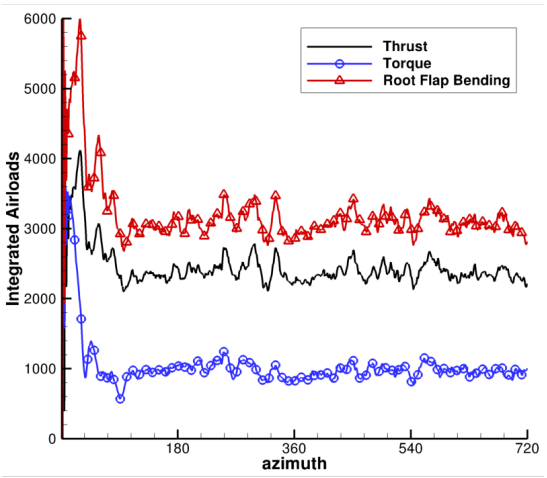
A summary of integrated rotor airloads for configurations simulated with near-body vorticity insertion accounting for bound vorticity is presented in Table 40. The corresponding time histories are presented in Fig. 125. For both the isolated rotor and the full wind turbine configurations, errors were reduced in the predicted out-of-plane forces and moments. 13% and 9% reductions in thrust and root flap bending error were observed in the case of the isolated rotor. In comparison, between 2–3% reductions in thrust and root flap bending error were observed for the full turbine. Torque predictions worsened by 2–4% for the two cases.

6.5.4 Performance

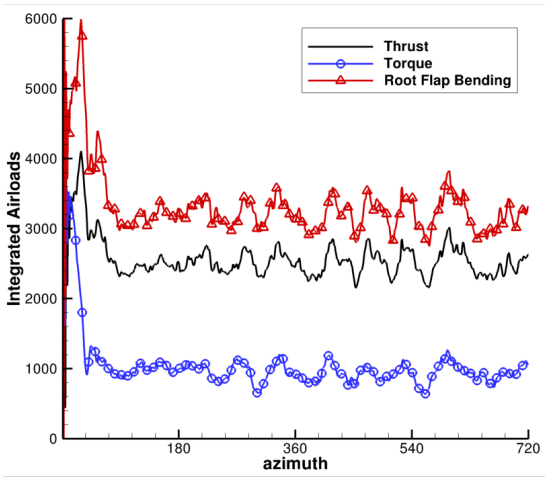
A comparison of computational cost for the traditional and hybrid URANS methodologies is presented in Table 41. Simulations were performed on an IBM iDataPlex cluster populated with 2.6 GHz Intel Xeon Sandy Bridge eight-core processors, typically with between 128 and 256 total processors. The cost of advancing the FUN3D near-body solution and VorTran-M wake solution have been separated, with

Table 40: Mean integrated loads (thrust, torque, and root flap bending, RFB) for hybrid calculations of the NREL VI isolated rotor and full horizontal-axis wind turbine (HAWT), with the inclusion of bound vorticity.

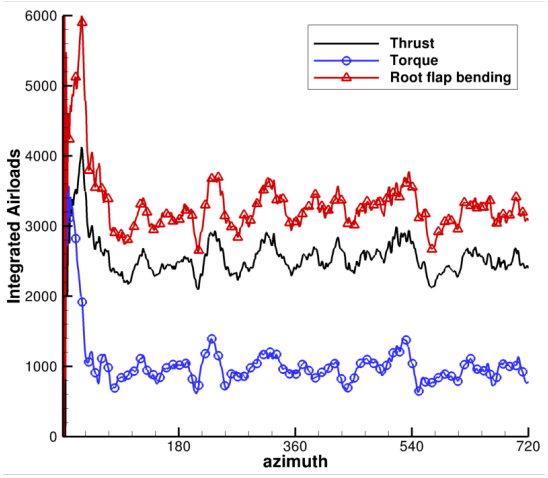
Case	Steps	Thrust [N]	Err. [%]	Torque [N*m]	Err. [%]	RFB [N*m]	Err. [%]
Experiment	–	2275	–	1104	–	3040	–
Rotor	1440	2365	3.82	968	-12.3	3077	1.22
Rotor, refined wake	1440	2494	9.48	932	-15.6	3173	4.38
HAWT	1440	2563	12.5	1047	-5.20	3307	8.77



(a) Isolated rotor



(b) Isolated rotor, refined wake



(c) Full wind turbine

Figure 125: Time history of integrated blade airloads for hybrid calculations including near-body vorticity insertion.

the FUN3D cost scaled by the number of processors utilized. The serial version of the VorTran-M wake module has been utilized with further cost reduction expected if executed in parallel. Since the wake domain increases in size as the wake is convected downstream, the cost of advancing the VorTran-M solution increases over time. The cost per step of Vortran-M presented in Table 41 is the wallclock time for the final simulation time step, which is an approximate upper bound on the cost of the wake solution. For the test cases evaluated, the cost of the wake solution is between two and three orders of magnitude less than the URANS near-body solution.

Table 41: Comparison of wallclock time to simulate the NREL Phase VI wind turbine (WT) for traditional URANS and hybrid URANS methodologies using baseline grids.

Case	FUN3D Parallel Cost/Step [s]	VorTran-M Cost/Step [s]
Overset Full WT	7.36×10^4	n/a
Hybrid Full WT	3.90×10^4	1.01×10^2
Overset Rotor	5.08×10^4	n/a
Hybrid Overset Rotor	2.56×10^4	5.55×10^1
Hybrid Axisymmetric Rotor	1.82×10^4	6.66×10^1

The cost of the hybrid solution to simulate the full NREL Phase VI wind turbine with the decoupled boundaries approach was nearly 50% faster in terms of wallclock time than the traditional overset solution including a background grid. If the tower and nacelle may be neglected, then a 65% reduction in cost within the original overset paradigm may be realized by reducing the extent of the background. In the case of axial flow, rotational symmetry may be assumed and the overset background mesh may be eliminated. This approach only calculates a single rotor blade within FUN3D and is the most cost-effective approach, achieving a reduction in wallclock time of over 75% for a 70% reduction in computational node count (Table 36).

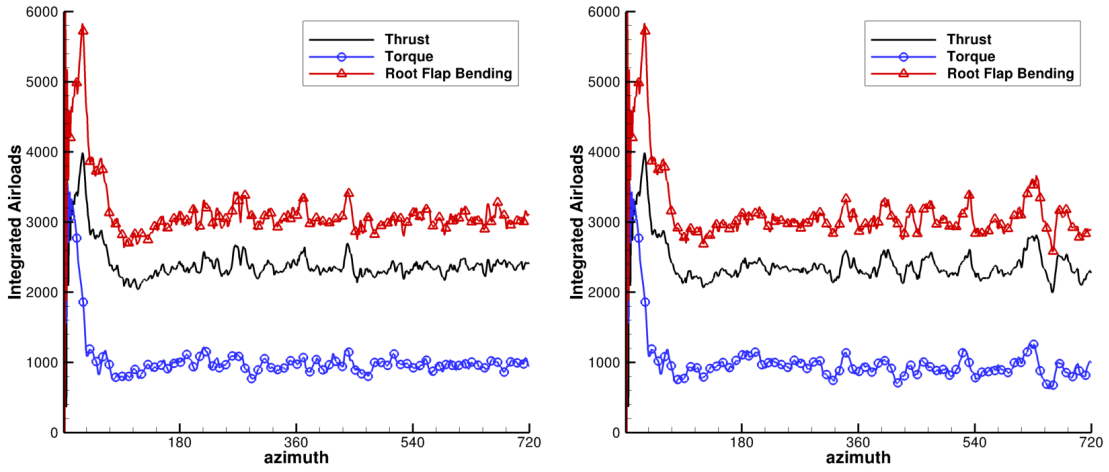
6.6 Application of Advanced Mapping Techniques

6.6.1 Isolated Wind Turbine Rotor

The thin-plate splines (TPS) and Wendland C^2 (W2) basis functions have been applied to interpolate near-body flow field data from FUN3D to the far-field wake solved by VorTran-M. At each solver iteration, velocities transferred by the RBF interpolation to the wake mesh determine the vorticity within each cell of the wake mesh which updates the wake solution. This updated solution modifies the near-body solution through the boundary condition as described in Section 6.1.2 for a fully-coupled implementation. All calculations include the insertion of bound vorticity.

All simulations were compared after two full rotor revolutions (Fig. 126). At this time, the oscillatory rotor loads have converged to a mean value with a standard deviation within 8% (of mean values) in thrust and root flap bending, and within 15% in torque (Table 42). The solution is numerically converged in all cases with solver residuals below 10^{-7}. Applying TPS interpolation reduced the amplitude of the unsteady oscillations (Fig. 126a) in comparison with the linear interpolation case (Fig. 125a). In contrast, W2 interpolation increased the amplitude of the unsteady loads (Fig. 126b) but the variations remained comparable to previous calculations for a variety of configurations (Table 39). Despite these different airloads histories, the integrated thrust and root flap bending values from all cases are within 1.5% of each other (Table 43). However, 5% additional error in the integrated torque was observed. The source of this error may be identified by considering the sectional airloads.

The blade pressure distributions for the different interpolation methods are plotted in Fig. 127. As with the other results in this chapter, the solution displays the most variability near to the blade root ($r/R = 0.30$). Increasing interpolation accuracy reduces suction near the leading edge and increases suction closer to the trailing edge, replicating the same trends as blade grid refinement and shifting the pressure distribution towards experimental values. This correction is also reflected in the



(a) Thin-plate spline interpolation

(b) Wendland C^2 interpolation

Figure 126: Time history of integrated blade airloads from an isolated rotor, calculated with RBF interpolated velocities (with bound vorticity inserted).

Table 42: Maximum standard deviation of converged rotor loads over the final partial revolution for calculations with linear and RBF interpolated velocities, presented as a percentage of the mean value.

	Thrust & Root Flap Bending	Torque
Linear interpolation	4.4%	7.5%
Thin-plate spline interpolation	2.9%	5.4%
Wendland C^2 interpolation	8.0%	17.2%

Table 43: Mean integrated loads (thrust, torque, and root flap bending, RFB) for hybrid calculations of the NREL VI isolated rotor, calculated with various interpolation techniques.

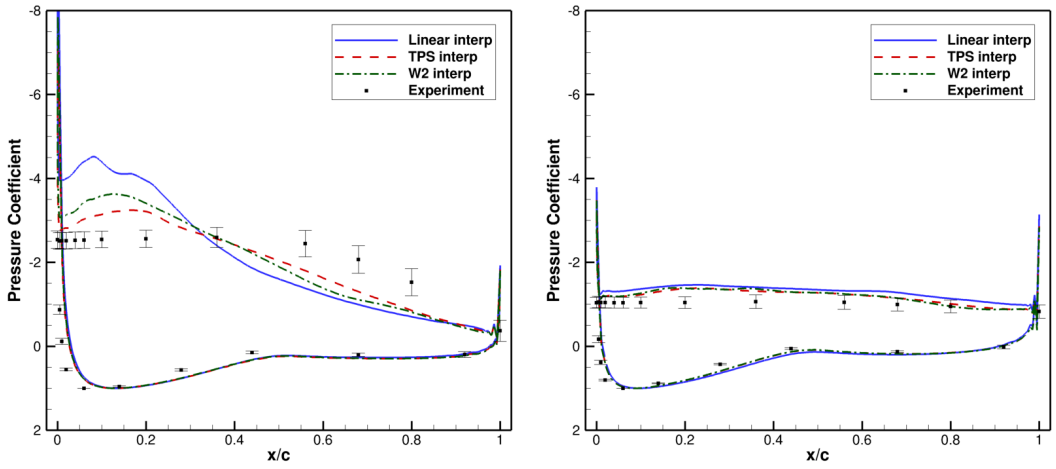
	Thrust [N]	Err. [%]	Torque [N*m]	Err. [%]	RFB [N*m]	Err. [%]
Experiment	2275	–	1104	–	3040	–
Linear interpolation	2365	3.82	968	-12.3	3077	1.22
Thin-plate spline interpolation	2371	4.08	961	-13.0	3061	0.69
Wendland C^2 interpolation	2361	3.64	914	-17.2	3032	-0.26

section tangent force and pitching moment distributions (Fig. 128b and c). Similar to the pressure distribution calculated with the refined blade (Fig. 109), the results applying RBF interpolation also have reduced suction near the trailing edge at $r/R = 0.47$, with reduction in pressure errors up to 20% (e.g., at $x/c = 0.80$). Further outboard, the W2 and TPS interpolation methods are closer to the experimental pressure distribution at the $r/R = 0.63$ and 0.80 radial stations, respectively. At $r/R = 0.63$, the error in pressure is reduced by about 15% near mid-chord (at $x/c = 0.56$) when interpolating with the W2 function. At $r/R = 0.80$, error is reduced by about 10% near the leading edge (at $x/c = 0.06$) when applying the TPS function. These differences translate into a quantitative improvement (by 8%) in the normal force prediction at $r/R = 0.63$ with W2, and a qualitative improvement in the tangent force prediction at $r/R = 0.80$ with TPS (Fig. 128a).

Considering the integrated torque predicted by the TPS calculation which was nearly identical to the linear calculation (Table 43), the increase in tangent force at $r/R = 0.80$ is offset by a decrease in tangent force at $r/R = 0.30$. In comparison, the decrease in tangent force near the root ($r/R = 0.30$) predicted by the W2 calculation is matched by a decrease in tangent force closer to the tip at $r/R = 0.80$. While both of these corrections tend toward experiment, the errors at the root and tip no longer offset each other and therefore increase error in the integrated torque. Overall however, application of an RBF interpolation technique resulted in improvements over linear interpolation ($\sim 10\%$ error reduction) in predicted inboard pressure distribution as well as sectional force and moment.

6.6.2 RBF Interface Cost

The converged hybrid isolated rotor solution from Section 6.6.1 was restarted with varying numbers of processors (64, 128, and 256) to analyze the cost of RBF interpolation within an unsteady flow simulation. For the implementation of RBF velocity

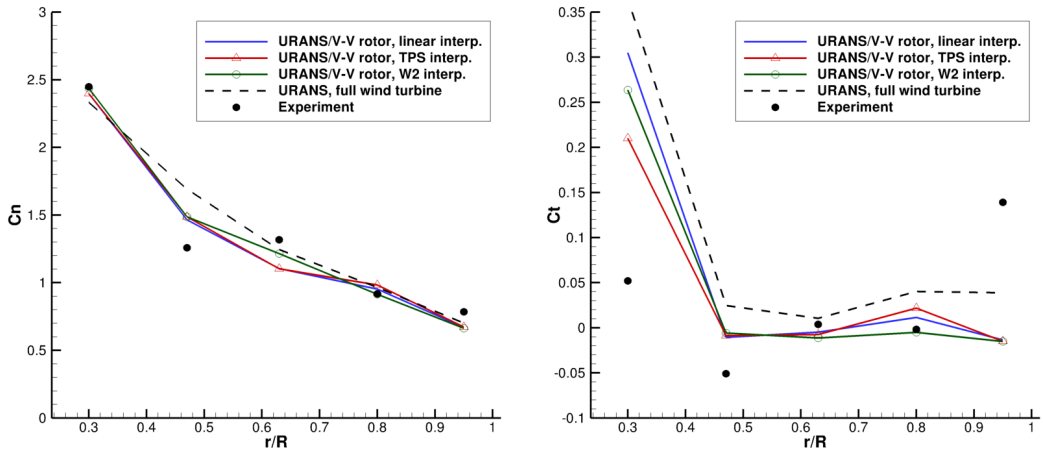


(a) $r/R = 0.30$ (b) $r/R = 0.47$

(c) $r/R = 0.63$ (d) $r/R = 0.80$

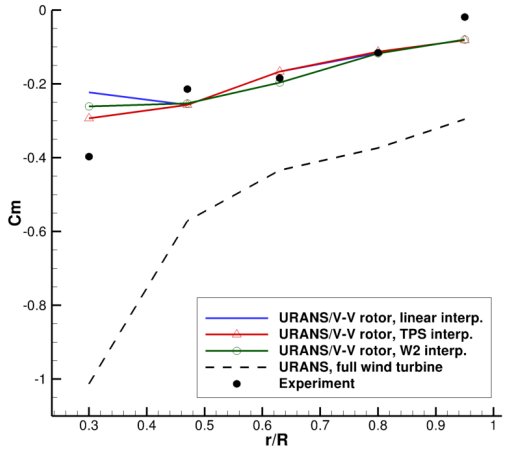
(e) $r/R = 0.95$

Figure 127: Comparison of pressure coefficients at radial stations corresponding to experimental measurements for a hybrid isolated rotor, calculated with linear and RBF interpolated wake velocities.



(a) Normal force coefficient

(b) Tangent force coefficient



(c) Pitching moment coefficient

Figure 128: Sectional airloads for hybrid isolated rotor calculations with linear and RBF interpolated wake velocities.

Table 44: Average CPU times in seconds for the k-d tree search compared with the average time to advance the near-body URANS solution.

Processors	Average nodes per processor	k-d tree creation [s]	Nearest node search [s]	URANS time
64	70,472	0.1620	238.1	227.1
128	35,236	0.0830	74.6	118.4
256	17,618	0.0320	23.4	68.9

interpolation described in Section 6.1.1, a k-d tree must be populated with the locations of all near-body grid nodes to enable a rapid search algorithm for finding the nearest node to a target location. Since nodal positions are not constant due to mesh motion, this k-d tree must be updated at each solver iteration. The most straightforward approach (which has been implemented) is to recreate the k-d tree after every iteration. The cost of creating a new tree was relatively inexpensive compared to the cost of descending through the tree to search for a nearest node (Table 44). The average tree creation CPU time scaled approximately linearly, decreasing with the number of processors. In contrast, the search time did not scale linearly, requiring significantly more time to descend through larger data trees. Even though the k-d tree algorithm can be very efficient, the search is still performed for all wake nodes which number over 2×10^5 within the current configuration. In the 64 processor case (containing the most nodes per processor on average), the search time for all wake nodes was comparable to the time to advance the near-body URANS solver suggesting that a different search algorithm should be applied.

The original interface applying linear interpolation loops over all cells in the near-body grid rather than all nodes on the far-field wake grid. The search routine terminates after locating a wake node contained by each near-body cell. A cell is determined to enclose a wake node when the barycentric mapping coordinates of the wake node

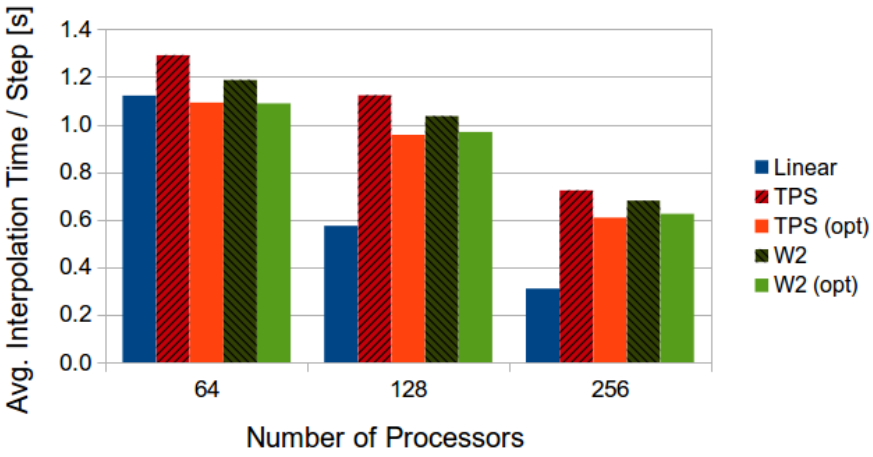


Figure 129: Average CPU time to perform interpolation at each solver time step for varying numbers of processors.

are all positive; when a cell is found, the linear interpolation weights are automatically known from the mapping coordinates (Section 3.1.3). This approach scaled linearly with the number of processors (Fig. 129), where the linear interpolation time included both the search and interpolation.

The CPU timings for RBF interpolation presented in Fig. 129 correspond to values in Table 45) and do not include the cost of the search procedure. Timings typically varied $\mathcal{O}(0.01)$ seconds between iterations so the times presented were averaged over only five iterations. Between 93–95% of the observed interpolation cost was attributed to the actual solution of the interpolation problem (Eqn. 28, in Chapter 3), with the additional overhead associated with gathering additional interpolation source points surrounding each nearest node. The cost of an initial naive implementation is presented along with an optimized implementation (labeled "opt" in Fig. 129). The naive implementation calculates the full left-hand side matrix while the optimized implementation assumes a symmetric matrix which is valid for RBF centers coincident with the locations of sampled data (Section 3.1.5). The lower triangular portion of the matrix is calculated and entries are mirrored across the diagonal. In addition,

Table 45: Cost to perform interpolation with different methods in terms of CPU time (seconds) and the fraction of the time to advance one iteration of the near-body URANS solver.

	Processors	Interpolation time [s]	Fraction of URANS time [%]
Linear	64	1.1214	0.50%
	128	0.5740	0.48%
	256	0.3100	0.44%
TPS	64	1.3014	0.58%
	128	1.1294	0.97%
	256	0.7340	1.09%
TPS (optimized)	64	1.0821	0.49%
	128	0.9521	0.83%
	256	0.6154	0.91%
W2	64	1.1980	0.52%
	128	1.0414	0.86%
	256	0.6834	0.98%
W2 (optimized)	64	1.0974	0.48%
	128	0.9581	0.80%
	256	0.6154	0.88%

the optimized implementation evaluates the TPS and W2 functions as follows:

$$\text{set } \rho = r^2; \qquad \phi(r) = \frac{1}{2}\rho \log(\rho). \tag{94}$$

This form of the thin-plate spline takes advantage of a mathematical property of logarithms and avoids a square root operation.

$$\text{set } \rho = 1 - r; \quad \text{set } \rho = \rho * \rho; \quad \text{set } \rho = \rho * \rho; \qquad \phi(r) = \rho(4r + 1) \tag{95}$$

Depending on compiler optimization, this form of the Wendland C^2 function accelerates evaluation of the $(1 - r)^4$ term.

For TPS interpolation in the current problem, taking advantage of the interpolation matrix symmetry reduced cost by 13–14% for all numbers of processors utilized, and applying Eqn. 94 further reduces cost by 3–4%. In comparison, the W2 function is initially less expensive to evaluate due to the definition of compact support (Section 3.1.5) and the absence of the logarithm function. Improvements of 8–9% in cost were observed by taking advantage of symmetry but applying Eqn. 95 resulted in a speed-up of at most $\sim 1\%$. After optimization, the cost for RBF interpolation with either function was nearly equal (Fig. 129).

In comparison with linear interpolation, the observed cost of performing RBF interpolations did not scale linearly (Fig. 129). At the smallest number of processors tested, the interpolation cost is comparable between both approaches. However, while the cost increases with decreasing number of processors, the fraction of time spent performing the interpolation constitutes a smaller portion of the time spent by the flow solver to advance the URANS solution (Table 45). For all cases tested, the cost of interpolation alone was less than 1.1% of the average URANS cost per step. Since the k-d search was relatively expensive, a more realistic estimate of cost was obtained by assuming that the search time is equal to the cost of the linear interpolation, which locates cells while calculating mapping coordinates. The combined linear search time

and RBF interpolation time still constitutes a small portion of the URANS cost per step (Table 44), between 1.0–1.4%.

6.6.3 Alternative Hybrid Interface Analysis

The hybrid approach applied in this thesis (detailed in Section 6.1) requires an interpolation step to collocate the near-field solution points (from an unstructured mesh) to far field points (on a Cartesian mesh), followed by a finite difference step to approximate the numerical gradients of velocity. The vorticity field passed to the wake solver is therefore subject to both interpolation and discretization errors. As an alternative, the RBF interpolants offer a continuously differentiable representation of the solution space, so it is possible to forgo the latter step and completely eliminate discretization errors from the solution.

The approximate derivative is derived from the RBF interpolant described in Section 3.1.5,

$$s_{f,X}(x) = \sum_{j=1}^{N} \alpha_j \Phi(x - x_j) + \sum_{k=1}^{Q} \beta_k p_k(x). \tag{96}$$

Differentiating this equation in space yields

$$\frac{\partial s}{\partial x_i} = \frac{\partial r}{\partial x_i} \sum_{j=1}^{N} \alpha_j \frac{\partial \phi(r)}{\partial r} + \sum_{k=1}^{Q} \beta_k \frac{\partial p_k(x)}{\partial x_i}, \tag{97}$$

where α_j and β_k are known from solving a system of interpolation equations (Eqn. 28). Any scaling of the local interpolation region is reflected in a normalization factor within $\dfrac{\partial r}{\partial x_i}$. Considering the same basis functions as in Section 3.4, the RBF gradient can be analytically expressed as:

$$\frac{\partial \phi}{\partial r} = r(1 + 2\log r) \tag{98}$$

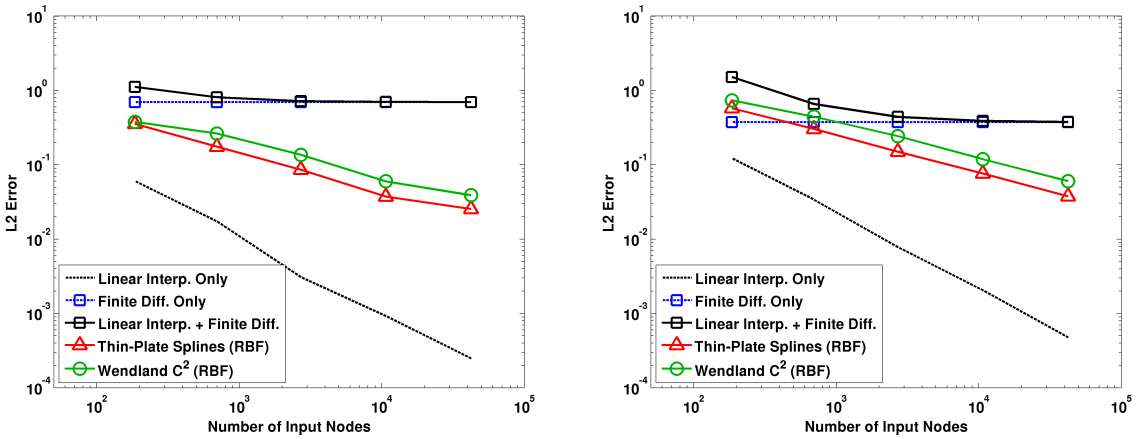
for the TPS basis, and

$$\frac{\partial \phi}{\partial r} = \frac{20r}{4r^2 - 3r - 1}\phi(r) = 20r(r-1)^3 \tag{99}$$

for the W2 basis.

A numerical experiment was conducted to transfer function values from a relatively fine unstructured input mesh into gradients on a coarse Cartesian mesh. The cell velocity gradients are then calculated as an average of first-order finite differences between cell nodes. Alternatively, an RBF mapping between the near-body and background meshes (which provides interpolation coefficients α and β) automatically estimates the gradient at the cell centers when applying Eqn. 97 with either Eqn. 98 or 99. This computational setup reflects the original FUN3D/VorTran-M hybrid interface methodology in which velocities are interpolated from unstructured near-body meshes onto the nodes of a Cartesian background mesh. Two background mesh sizes of $\Delta s = 0.1$ and 0.05 have been considered, corresponding to an 11×11 and 21×21 grid on a unit domain, $[0\ 1]^2$. The unstructured input meshes are generated with the `MATLAB initmesh` and `refinemesh` codes, producing isotropic meshes with $324{\times}4^N = 324$, 1296, 5184,... cells for grid levels $N = 0, 1, 2, ...$ For $N = 0$, the approximate cell size is on the same order as the fine background mesh spacing, with $\Delta s \approx 0.056$. For $N = 4$, $\Delta s \approx 0.0035$ and is two orders of magnitude smaller than the coarse background mesh.

A comparison of errors in estimating the gradients of Franke's analytical function (Eqn. 35) is plotted in Fig. 130, with the sources of error decomposed into interpolation and finite-difference components. Only the gradients in the x direction have been shown as the errors in the x- and y-directions are indistinguishable. The "Linear Interp. Only" curve (Fig. 130) indicates the interpolation error accrued in the first part of the data transfer. This error decreases linearly with an expected slope of approximately negative one (-1.03), verifying order accuracy. To isolate the discretization error, finite difference approximations were applied on the background mesh starting with the exact solution at the nodes rather than interpolated values. The "Finite Diff. Only" curve then provides an estimate of the isolated discretization error, bypassing the interpolation step and beginning with the exact solution at the



(a) 11×11 Output Mesh (b) 21×21 Output Mesh

Figure 130: Comparison of errors in estimating gradients from Franke's function (Eqn. 35). Linear interpolation error is compared with the exact function while all other curves are compared with the exact function gradient.

background mesh nodes. This constitutes the second part of the data transfer. The coarse mesh discretization error (Fig. 130a) is approximately double the fine mesh discretization error (differing by a factor of 1.86), corresponding to a doubling of mesh size on the coarse mesh.

When the two steps are combined, the combined interpolation and discretization errors do not form a simple sum. Instead, the combined errors are approximately an order of magnitude higher than the linear interpolation alone for the coarsest mesh tested. As the input mesh is refined, the interpolation errors reduce further and the combined gradient estimation error approaches the constant, output mesh-dependent finite difference result.

Under the same grid configurations, Figure 130 indicates that the RBF-based gradient estimates start at nearly the same order of magnitude as the ideal finite-difference estimate or lower, depending on the mesh density. In all cases, the RBF-based gradients are less than the standard interpolation and finite difference estimate. The RBF approach has no dependence on the output mesh size, unlike the standard approach. Therefore as the input mesh is further refined, the RBF mapping estimate

continues to improve, albeit at a slower than linear rate (with a slope of approximately 0.5). Both RBF methods converge at the same rate, with the TPS basis consistently outperforming the compactly-supported Wendland C^2 function.

CHAPTER VII

CONCLUSIONS

This thesis has investigated a variety of modular data transfer strategies for complex flow simulation applicable to both overset and hybrid methodologies. This research provides two contributions to the computational community:

- A practical algorithm for orphan-free overset data transfer was developed based on scattered data interpolation techniques and applied to test problems including a convecting vortex, shock tube, and ship airwake. These simulations demonstrate applicability to a variety of flow conditions, including inviscid and viscous flows, both with and without discontinuities. In the presence of orphans under the traditional overset paradigm, this new algorithm successfully preserved the structure of a convecting vortex and reduced transient conservation errors by an order of magnitude.

- A novel steering strategy has been developed and applied as a data-adaptive interpolation algorithm on two-dimensional solution fields. The method provides variable resolution depending on gradients of the solution field and is a scattered-data analog to a slope limiter. In addition to increasing interpolation accuracy in the presence of discontinuities, numerical stability is also enhanced.

These techniques are decoupled from solver type (URANS, vorticity-velocity, potential flow) and topology (Cartesian, structured, and unstructured) and may be readily applied within existing methodologies. To demonstrate the practicality of this approach, the original scattered data interpolation has also been applied to transfer the unsteady near-body velocity field within a fully-coupled hybrid URANS/vorticity-velocity methodology.

To provide a basis for the data transfer algorithms developed in this thesis, scattered data interpolation techniques have been evaluated for both smooth and discontinuous solutions. These techniques include radial basis function (RBF) interpolation with two different basis functions, the globally supported thin-plate spline (TPS) and the compactly-supported Wendland C^2 function, in addition to the established inverse isoparametric mapping (IIM) technique. The following observations have been made:

- A robust data transfer technique should be able to handle arbitrarily distributed data in any dimension. Both the RBF and IIM techniques are able to accommodate clouds of source points. These clouds have been categorized into *local*, *neighborhood*, and *global* distributions.

- State-of-the-art data transfer codes employ linear mappings which depend only upon the set of source points provided by the nodes of the computational cell (or element) enclosing the target point. For this *local* approach, application of higher-order approaches offered no improvements in accuracy. The number of data points utilized in a linear interpolation should be considered the minimum amount of information to support a higher-order interpolation technique.

- Additional points may be sourced from surrounding nodes or cells to take advantage of pre-existing connectivity information. In this *neighborhood* approach, approximately an order of magnitude reduction in error was observed for fine-to-coarse interpolation of a smooth solution field with the RBF approaches. The cloud included sixteen nodes, formed by including one additional level of nodes surrounding each target point. When the cloud was expanded to include two levels of surrounding nodes (36 nodes), the rate of convergence increased from second-order to approximately $\mathcal{O}(\Delta s^{2.5})$ where Δs is the uniform grid spacing. In comparison, a higher-order six-node IIM provided the same level of accuracy

as the 36-node RBF, despite dispersive error observed at coarser grid levels; this IIM approach converged at $\mathcal{O}(\Delta s^{2.8})$.

- The RBF technique also permits straightforward application of a *global* interpolation cloud, in which all target points are simultaneously solved by considering all available source points. This approach achieved the highest rate of convergence, with nearly two orders of magnitude reduction in error over linear techniques, and achieved the fastest convergence of up to $\mathcal{O}(\Delta s^{3.5})$. However, this approach quickly became prohibitively expensive in terms of memory requirements for even a two-dimensional 161×161 point mesh.

- A neighborhood RBF approach is recommended since it offers reasonable accuracy and may be tailored for specific problems, grid configurations, and accuracy/efficiency requirements. Accuracy and convergence order are increased by straightforward addition of source points.

- The inverse isoparametric mapping (IIM) approach provides linear mappings in state-of-the-art computational applications but can be applied with higher-order elements to permit the interpolation of scattered data. Evaluation of IIM with quadratic triangle elements demonstrated both accuracy and efficiency (in the number of required source points) when interpolating an analytical PDE solution and a blade surface pressure solution on structured grids. However, for arbitrarily distributed data, the isoparametric mapping can become non-unique, which renders its application to unstructured grids and/or grids with orphan points not straightforward.

- Under repeated applications, all higher-order, neighborhood-based interpolation approaches (RBF and IIM) diverged from the starting solution due to eventual accumulation of non-dissipative errors.

Typical aerodynamic problems of interest can feature discontinuous solution quantities, either due to deterministic behavior (compressible flow) or stochastic behavior (turbulent fluctuations). This motivates application of the data-adaptive steering technique. Both an analytical step function and a ship airwake solution were investigated with the following results:

- For a two-dimensional step function, interpolation with the thin-plate spline or Wendland C^2 functions provided error reductions of up to 26% over linear interpolation. In comparison, application of the steered Wendland C^2 approach provides error reductions of nearly 60% compared to bilinear interpolation. The steering approach was demonstrated to be an effective scattered-data limiter, able to mitigate the Gibbs phenomenon with solution overshoot reduced from 12% to 1%.

- The steering technique is able to increase numerical stability by decreasing the condition number of the interpolation matrix to be solved.

- Interpolation of a ship airwake solution using RBF approaches (steered and unsteered) reduced errors in both mean-flow and turbulence variables by approximately half an order of magnitude over linear interpolation at the finest grid levels tested. All RBF results were within 10% of each other. When considering the gradients of the mean-flow and turbulence variables, the steered Wendland C^2 approach stood out as the only method that consistently outperformed linear interpolation, with between half to a full order of magnitude reduction in error.

- In general, solution fields with discontinuities exhibited slower rates of convergence. For the two-dimensional step function, both linear and RBF interpolations were non-convergent for increasing wake refinement, with an initial slope of approximately 0.6 and a final slope of zero. In contrast, the steered

Wendland C^2 was able to maintain convergent behavior, with an initial nearly second-order rate of 1.9 and a final rate of 0.5. Application of RBF techniques to the ship airwake problem accelerated convergence by 5–49% over the linear approach in mean-flow and turbulence variables. The highest rates of convergence observed ($\mathcal{O}(\Delta s^{1.4})$) were obtained with the steered interpolation. Higher than second-order rates of convergence were not achieved by any of the other approaches considered.

The remainder of the thesis evaluates standard (unsteered) RBF interpolation in two and three dimensions within fluid dynamics solvers. The work focuses on determining the applicability of the scattered data interpolation approach to overset and hybrid methodologies and assessing its impact on accuracy, rather than optimizing for accuracy or efficiency. In this assessment, the neighborhood approach was applied with additional points sourced from a single level of neighbors. Resulting interpolation clouds contained on average 14–24 source points in two dimensions and 32 source points in three dimensions.

Two-dimensional overset URANS simulations of a convecting vortex (with and without orphan points) and a shock tube were performed to validate the RBF approach.

- For the convecting vortex test case, overset errors were isolated from temporal and spatial errors by Richardson extrapolation. In configurations without orphan points, error reductions of up to 72% over standard linear implementation were observed when applying thin-plate splines RBF interpolation, and up to an order of magnitude reduction when applying Wendland C^2 interpolation.

- For overset configurations without orphan points, the overset interpolation error constitutes only 10–13% of the spatial errors for the standard overset method at the baseline level. Since dissipative error is reduced with grid refinement, a

greater portion of the spatial error (11-18%) originates from overset interpolation at the fine grid level. In this case, application of either thin-plate splines or the Wendland C^2 basis can reduce spatial errors to within 8% and 6% of the single grid solution for less than 3% increase in cost.

- The largest improvements when applying RBF over linear interpolation were observed when calculations included orphan points. The increase in accuracy comes from the ability to perform extrapolation. In the baseline configuration with orphans, half an order of magnitude reduction in streamwise velocity errors and over an order of magnitude reduction in pressure errors were observed. On the refined grid, RBF interpolation was able to virtually eliminate all orphan-related overset errors. The thin-plate splines overset scheme reduced the spatial errors of all variables to within 8% of a standard orphan-free solution employing linear interpolation; the Wendland C^2 scheme reduced all errors to within 11%.

- The most pronounced reduction in isolated overset error was achieved with RBF overset interpolation and observed in the streamwise velocity. On the coarsest grid configuration with orphans (coarser than recommended for CFD), overset errors were less sensitive to interpolation technique. For instance, the streamwise velocity errors were reduced by 73–76% when applying either thin-plate splines or the Wendland C^2 function at the baseline and fine grid levels; in comparison, the overset error reduction was less than half as effective (30-33%) indicating on coarser grids the spatial errors tend to be dominated by dissipative error rather than overset interpolation error.

- As with the rotor blade test solution, repeated interpolations on a degraded solution can cause the solution to diverge. On the coarse grid configuration with orphans, the pressure solution was unstable when applying RBF interpolation. This matches trends observed when studying repeated interpolations of

a rotor blade solution and a step function solution in which the Wendland C^2 interpolation was the most unstable when solution quality was poor.

- RBF interpolation also reduces transient errors in addition to spatial errors, making it a viable option for addressing orphan points. Vortex passage through overset boundaries with orphans caused perturbations to the fluxes on the outer boundary of the composite computational domain. The RBF methods reduced the magnitude of these conservation errors by up to an order of magnitude and more quickly attenuated transient errors after the vortex passed through the overset boundary. In comparison, linear overset interpolation generated conservation errors that persisted even after the vortex core had completely passed through the overset fringe region. These errors persisted in the streamwise momentum for the remainder of the simulation.

- When orphan points were present, the increase in cost over a standard overset implementation is more significant (up to 12% and 16% on baseline and fine grids) than the orphan-free configuration, due to a four-fold increase in subiterations performed. However, this increase in cost can be kept in check by parallel code execution which would distribute the RBF interpolation operations across multiple processors. For comparison, overset solution of a three-dimensional ship airwake contained an order of magnitude more fringe points but only incurred a 6% increase in cost per step.

- As an alternative to applying the steering strategy, a more traditional limiter may be imposed by prohibiting the creation of new extrema. This has been demonstrated to be effective at stabilizing the solution of a moving shock. For a shock tube problem, solution integrity was maintained with up to 2.6% error reduction in instantaneous pressure errors. Where the moving shock spanned multiple overset grids, over 10% reduction of temperature error was observed

with the RBF methods compared to the linear overset implementation.

The RBF overset interface was further extended to handle three-dimensional test cases and a turbulent ship airwake simulation was performed to study RBF performance with three-dimensional, turbulent flows. The following observations have been made:

- The ship airwake was characterized by unsteady, separated flow most notably in the hangar region. Even though the overset boundaries at which interpolations were performed were located approximately half a ship length away from the ship (at hangar height), the overset flow solution was sensitive to the choice of interpolation method in terms of flow-field features, sampled velocities, wake spectra, and flow separation behavior on the deck.

- In comparison with experiment, the averaged flow characteristics above the flight deck were comparable to experiment. The sampled velocity magnitudes for all approaches differed from experimental measurements by at most 4% of the free-stream velocity, and the flow trajectories (in terms of local flow pitch and yaw angles) were within 2° of experiment.

- In comparison with a single grid calculation, the RBF overset results had 7% less error in flow separation location (on the flight deck) than the linear overset solution, differing from the single grid location by 2%. Wake turbulent kinetic energy spectra calculated by the RBF overset approaches were closer to the single grid calculation than the linear overset result, with the thin-plate splines solution having 9% less error on average and the Wendland C^2 solution reducing the maximum error by up to 20%.

An efficient hybrid methodology coupling a URANS analysis with a vorticity-velocity analysis was improved to more efficiently model a variety of helicopter and wind turbine geometries under axisymmetric and decoupled boundary configurations.

This approach has been validated and demonstrated for the analysis of NREL Phase VI wind turbine components (including tower, isolated rotor, and full turbine configurations). In summary:

- The hybrid solutions of an isolated, axisymmetric rotor were within 11% of integrated thrust and root flap bending predicted by a traditional URANS simulation of a full turbine. A 2% improvement in predicted torque in comparison with experiment was observed. Moreover, results for this optimized grid configuration were obtained with 75% less wallclock time.

- A decoupled boundaries strategy for unstructured grids was developed and applied to the full NREL horizontal axis wind turbine configuration, permitting non-contiguous grids to be solved within the same hybrid framework. Predictions were within 9% of the integrated thrust and root flap bending computed by a URANS-alone methodology, demonstrating a 7% improvement in integrated torque predictions compared with experiment. These results were obtained in nearly 50% less wallclock time.

- Bluff body physics of the wind turbine tower demonstrated excellent correlation with an analogous cylinder in crossflow in terms of drag, shedding frequency, and separation location. All three quantities computed by the hybrid methodology were within experimental bounds and within 7% of an identical standalone URANS analysis.

- Results obtained with the hybrid methodology were compared to findings from an NREL blind study. Section normal force and pitching moment predictions were closer to experiment than other researchers in the blind study. In addition, the hybrid pitching moment predictions were a marked improvement over the standalone URANS results, which over-predicted the magnitude of the pitching moment by a factor of two to three.

- Stalled flow at the blade tip indicated by sectional pressure distributions had an insignificant effect on predicted normal force as evidenced by section normal force distributions and the integrated thrust. However, this difference from experiment resulted in a significant reduction in the tangential load carried by the blade tip. This reduction in the blade tip load is offset by an increase in the tangential loads near the blade root, highlighting the challenges inherent in the analysis of separated flow.

The hybrid rotor simulation allows demonstration of repeated RBF interpolation in an unsteady, moving mesh application. Findings include:

- The fully-coupled hybrid solution was sensitive to variations in the near-body pressure and velocity field due to wake refinement, near-body grid adaptation, or chosen interpolation method. These sensitivities were observed in the blade pressure and sectional load distributions; however the integrated effect of all these variations was similar, with errors from all hybrid rotor simulations within 17% of experiment in thrust, torque, and root flap bending.

- Increasing interpolation accuracy with RBF interpolation produces trends near the blade root comparable to near-body grid refinement. While grid refinement is performed to increase flow resolution, it has the added benefit of improving data transfer accuracy. Therefore simply improving interpolation accuracy is expected to have a similar effect on the coupled hybrid solution. Improvements to pressure distributions were most notable near the blade root at 30% span, with corrections on the order of 10%. These changes in the pressure distribution are reflected in the section tangent force and pitching moments which tend toward experiment near the blade root.

- Applying RBF interpolation within the hybrid interface is less scalable than linear interpolation in terms of cost. However, the increase in CPU time during

each iteration is estimated to be less than 1.5% for up to 256 processors.

7.1 Future Work

Based on the success of the orphan-free overset implementation with radial basis functions, future work should include:

- The overset interface may be optimized to include mixed interpolation approaches. Within this thesis, the same interpolation method was used throughout the flow field for consistency. However, for efficiency a linear interpolation may be utilized unless there are orphan points. Alternatively, inverse isoparametric mapping with higher-order elements may also be applied instead of linear interpolation. Applying mixed interpolation methods requires special treatment at locations where different interpolation methods meet. One approach is to overlap the different interpolation regions and then average the results from different methods to guarantee a smooth transition of the flow field [73].

- The overset interface should be tested with moving grids. This will require dynamic update of the overset assembly, therefore the interpolation matrices for all donor-receptor sets may not be solved ahead of time. Efficiency gains when performing a dynamic overset solution may be achieved by running in parallel and having a single processor dedicated to solving the RBF interpolation system. This may be the same processor that updates the overset connectivity information "on the fly." Afterward, more complex overset demonstration cases involving both orphan points and moving meshes should be evaluated. A natural application of this solution methodology is to the analysis of store separation which frequently encounters orphans [8].

- A feature of the compactly-supported Wendland C^2 basis function that has not been exploited is the fact that the function decays to exactly zero. Therefore,

a neighborhood of interpolation points may be expanded to include additional donors while keeping the support radius fixed. As a result, this will increase resolution due to the inclusion of additional source points, but it will also guarantee sparsity in the interpolation matrix (Eqn. 28). It will then be possible to take advantage of sparse matrix solution algorithms to increase computational efficiency.

- Since the RBF approach is directly extensible to arbitrarily high dimensions, a fourth-dimension may be added to perform ensemble interpolations in both space and time.

Further work on the application of radial basis functions to computational data transfer is recommended:

- A consistent and efficient implementation of a steered radial basis function overset methodology motivates closer integration of the donor/receptor transaction library and the flow solver. For example, the solution gradients already calculated by the solver (which may employ more sophisticated methods of gradient estimation [118]) may be applied to steering the interpolation rather than naive calculation of gradients from sampled velocities. Therefore, an enhanced overset framework should be developed that permits direct access to flow-field gradients calculated by the flow solver. This can provide a more accurate description of the flow field to facilitate the application of steered interpolation strategies. This will also eliminate uncertainty in the gradient estimation part of the steering algorithm.

- The current two-dimensional implementation of the steering kernel regression should be extended to three dimensions. This may be based on the work of Takeda et al. [114], who applied the approach to the interpolation of video which includes two spatial dimensions and a temporal dimension. After the

effectiveness of the technique is verified, an identical RBF steering procedure may be applied in which the three-dimensional steering parameters are applied to transform the RBF interpolation space.

- Since an RBF provides a continuous, differentiable mapping of a solution field, it is possible to apply the RBF approach to interpolate the near-body velocity field and directly recover the corresponding vorticity field. The accuracy of such an approach in practice should be assessed in addition to numerical stability.

One direct application of the present work is grid adaption, which seeks to strategically insert, move, and/or remove points in a computational mesh. Grid adaption has the potential to significantly increase resolvability, as well as increase efficiency by clustering points in regions where they are most needed to capture flow-field features of importance. An excellent review of approaches and advancements to the current grid adaption state of the art are presented in Ref. 110. When applying this technique points are typically added to the flow field. However, to enable time-accurate adaptation, new points must be given a solution value through interpolation to permit restart of the flow solution. Therefore maintaining high solution accuracy at the new points is necessary as to not introduce transient errors into the solution.

Bibliography

[1] Benioff, M. R. and Lazowska, E. D., "Computational Science: Ensuring America's Competitiveness," Tech. Rep. PITAC, President's Information Technology Advisory Committee, June 2005.

[2] Komerath, N., Smith, M. J., and Tung, C., "A Review of Rotor Wake Physics and Modeling," *Journal of the American Helicopter Society*, vol. 56, no. 2, pp. 022006/1–19, 2011.

[3] Steger, J., Dougherty, F., and Benek, J., "A Chimera Grid Scheme," *American Society of Mechanical Engineers, Fluids Engineering Division*, vol. 5, pp. 59–69, 1983.

[4] Benek, J., Steger, J., and Dougherty, F., "A Flexible Grid Embedding Technique with Application to the Euler Equations," in *6th AIAA Computational Fluid Dynamics Conference*, no. AIAA-83-1944, (Danvers, Massachusetts), July 13–15, 1983.

[5] Abras, J., Lynch, C. E., and Smith, M., "Computational Fluid Dynamics-Computational Structural Dynamics Rotor Coupling Using an Unstructured Reynolds-Averaged Navier-Stokes Methodology," *Journal of the American Helicopter Society*, vol. 57, pp. 1–14, Jan 2012.

[6] Quon, E., Smith, M. J., Whitehouse, G. W., and Wachspress, D. A., "Unsteady Reynolds-Averaged Navier-Stokes-Based Hybrid Methodologies for Rotor-Fuselage Interaction," *AIAA Journal of Aircraft*, vol. 49, pp. 961–965, May/June 2012. doi: 10.2514/1.C031578.

[7] Lynch, C. E. and Smith, M. J., "Unstructured Overset Incompressible Computational Fluid Dynamics for Unsteady Wind Turbine Simulations," *Wind Energy*, vol. 16, pp. 1033–1048, October 2013. Article first published online: 1 Aug 2012 — doi: 10.1002/we.1532.

[8] Power, G. D., Gudenkauf, J. A., Masters, J., Calahan, J. A., and Aboulmouna, M., "Integration of USM3D into the Store-Separation Process: Current Status and Future Direction," in *47th AIAA Aerospace Sciences Meeting*, (Orlando, Florida), AIAA-2009-339, January 5–8, 2009.

[9] Rogers, S. E., Suhs, N. E., and Dietz, W. E., "PEGASUS 5: An Automated Preprocessor for Overset-Grid Computational Fluid Dynamics," *AIAA Journal*, vol. 41, no. 6, pp. 1037–1045, 2003.

[10] Sitaraman, J., Floros, M., Wissink, A., and Potsdam, M., "Parallel Domain Connectivity Algorithm for Unsteady Flow," *Journal of Computational Physics*, vol. 229, no. 12, pp. 4703–4723, 2010.

[11] Noack, R. W., Boger, D. A., Kunz, R. F., and Garrica, P. M., "Suggar++: An Improved General Overset Grid Assembly Capability," in *19th AIAA Computational Fluid Dynamics Conference*, AIAA-2009-3992, 2009.

[12] Noack, R. W., "SUGGAR: A General Capability for Moving Body Overset Grid Assembly," in *17th AIAA Computational Fluid Dynamics Conference*, (Toronto, Ontario), AIAA-2005-5117, June 6–9, 2005.

[13] J. Guerrero, "Efficient Treatment of Complex Geometries and Moving Bodies using Overlapping Grids," in *10th ISGG Conference on Numerical Grid Generation*, (Crete, Greece), 2007.

[14] Press, W., Flannery, B., Teukolsky, S., and Vetterling, W., *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, ch. 2. New York, NY: Cambridge University Press, 2nd ed., 1992.

[15] Meakin, R. L., "On the Spatial and Temporal Accuracy of Overset Grid Methods for Moving Body Problems," in *12th AIAA Applied Aerodynamics Conference*, no. AIAA-94-1925, (Colorado Springs, Colorado), pp. 858–871, June 20–22, 1994.

[16] Noack, R. W., "Resolution Appropriate Overset Grid Assembly for Structured and Unstructured Grids," in *16th AIAA Computational Fluid Dynamics Conference*, (Orlando, Florida), AIAA-2003-4123, June 23–26, 2003.

[17] Gaitonde, D. and Visbal, M., "Padé-type Higher-Order Boundary Filters for the Navier-Stokes Equations," *AIAA Journal*, vol. 38, no. 11, pp. 2103–2112, 2000.

[18] Sherer, S. and Scott, J., "High-Order Compact Finite-Difference Methods on General Overset Grids," *Journal of Computational Physics*, vol. 210, no. 2, pp. 459–496, 2005.

[19] Noack, R. W., "DiRTlib: A Library to Add an Overset Capability to Your Flow Solver," in *17th AIAA Computational Fluid Dynamics Conference*, no. AIAA-2005-5116, (Toronto, Canada), June 6–9, 2006.

[20] Rogers, S. E., Roth, K., Nash, S. M., Baker, M. D., Slotnick, J. P., Whitlock, M., and Cao, H. V., "Advances in Overset CFD Processes Applied to Subsonic High-Lift Aircraft," in *18th AIAA Applied Aerodynamics Conference*, no. AIAA-2000-4216, (Denver, Colorado), August 14–17, 2000.

[21] Abras, J. N. and Hariharan, N., "CFD Solver Comparison of Low Mach Flow over the ROBIN Fuselage," in *AIAA Scitech 52nd Aerospace Sciences Meeting*, no. AIAA-2014-0752, (National Harbor, Maryland), January 13–17, 2014.

[22] Platt, S. C., "Parachute Extraction of a Generic Store from a C-130; a CFD Proof of Concept," Master's thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, March 2005. http://www.dtic.mil/docs/citations/ADA434226.

[23] Steger, J. L., "Thoughts on the Chimera Method of Simulation of Three-Dimensional Viscous Flow," in *Computational Fluid Dynamics Symposium on Aeropropulsion*, 1991.

[24] Alauzet, F. and Mehrenberger, M., "P1-Conservative Solution Interpolation On Unstructured Triangular Meshes," Tech. Rep. 6804, Institut National de Recherche en Informatique et en Automatique, 2009.

[25] Part-Enander, E. and Sjogreen, B., "Conservative and Non-Conservative Interpolation Between Overlapping Grids for Finite Volume Solutions of Hyperbolic Problems," *Computers and Fluids*, vol. 23, no. 3, pp. 551–574, 1994.

[26] Chesshire, G. and Henshaw, W., "Composite Overlapping Meshes for the Solution of Partial Differential Equations," *Journal of Computational Physics*, vol. 90, no. 1, pp. 1–64, 1990.

[27] Wang, Z., Hariharan, N., and Chen, R., "Recent Development on the Conservation Property of Chimera," *International Journal for Computational Fluid Dynamics*, vol. 15, pp. 265–278, 2001.

[28] Tang, H., Jones, S. C., and Sotiropoulos, F., "An Overset-Grid Method for 3D Unsteady Incompressible Flows," *Journal of Computational Physics*, vol. 191, pp. 567–600, 2003.

[29] Tang, H. and Zhou, T., "On Nonconservative Algorithms for Grid Interfaces," *Journal of Numerical Analysis*, vol. 37, no. 1, pp. 173–193, 2000.

[30] Chesshire, G. and Henshaw, W., "A Scheme for Conservative Interpolation on Overlapping Grids," *SIAM Journal of Scientific Computing*, vol. 15, no. 4, pp. 819–845, 1994.

[31] Zheng, Y. and Liou, M.-S., "A Novel Approach of Three-Dimensional Hybrid Grid Methodology: Part 2. Flow solution," *Computer Methods in Applied Mechanics and Engineering*, vol. 192, pp. 4147–4171, 2003.

[32] Freitas, C. and Runnels, S., "Simulation of Fluid-Structure Interaction Using Patched-Overset Grids," *Journal of Fluids and Structures*, vol. 13, no. 2, pp. 191–207, 1999.

[33] Quon, E., Smith, M., Whitehouse, G., and Wachspress, D., "Hierarchical Variable Fidelity Methods for Rotorcraft Aerodynamic Design and Analysis," in *Proceedings of the 67th American Helicopter Society Annual Forum*, (Virginia Beach, Virginia), May 3–5, 2011.

[34] Wachspress, D. A., Yu, K., Saberi, H., Hasbun, M. J., Ho, J. C., and Yeo, H., "Helicopter Rotor Airload Predictions with a Comprehensive Rotorcraft Analysis," in *Proceedings of the 68th American Helicopter Society Annual Forum*, (Fort Worth, Texas), May 1–3, 2012.

[35] Quon, E. W., Smith, M. J., and Whitehouse, G. R., "A Novel Computational Approach to Unsteady Aerodynamic and Aeroelastic Flow Simulation," in *International Forum on Aeroelasticity & Structural Dynamics*, (Bristol, United Kingdom), Royal Aeronautical Society, June 24–26 2013.

[36] Horn, J. F., Bridges, D. O., Wachspress, D. A., and Rani, S. L., "Implementation of a Free-Vortex Wake Model in Real-Time Simulation of Rotorcraft," *Journal of Aerospace Computing, Information, and Communication*, vol. 3, pp. 93–114, March 2006.

[37] D'Andrea, A., "Development of a Multi-Processor Unstructured Panel Code Coupled with a CVC Free Wake Model for Advanced Analyses of Rotorcrafts and Tiltrotors," in *Proceedings of the 64th American Helicopter Society Annual Forum*, (Montréal, Canada), April 29–May 1, 2008.

[38] Sitaraman, J. and Baeder, J. D., "Field Velocity Approach and Geometric Conservation Law for Unsteady Flow Simulations," *AIAA Journal*, vol. 44, pp. 2084–2094, September 2006.

[39] Whitehouse, G. R. and Boschitsch, A. H. and Smith, M. J. and Lynch, C. E. and Brown, R. E., "Investigation of Mixed Element Hybrid Grid-Based CFD Methods for Rotorcraft Flow Analysis," in *Proceedings of the 66th American Helicopter Society Annual Forum*, (Phoenix, Arizona), May 11–13, 2010.

[40] Smith, M. J., Lim, J. W., van der Wall, B. G., Baeder, J., Biedron, R. T., Boyd, D.D., J., Jayaraman, B., Jung, S., and Min, B.-Y., "The HART II International Workshop: An Assessment of the State-of-the-Art in CFD/CSD Prediction," *CEAS The Aeronautical Journal*, vol. 4, pp. 345–372, December 2013. DOI 10.1007/s13272-013-0078-8.

[41] Shih, T. I.-P., "Overset Grids: Fundamentals and Practical Issues," in *20th Applied Aerodynamics Conference*, no. AIAA-2002-3259, (St. Louis, Missouri), June 24–26, 2002.

[42] Marshall, D., *Extending the Functionalities of Cartesian Grid Solvers: Viscous Effects Modeling and MPI Parallelization*. PhD thesis, Georgia Institute of Technology, Atlanta, Georgia, 2002. `https://smartech.gatech.edu/handle/1853/11999`.

[43] Nichols, R. H., Tramel, R. W., and Buning, P. G., "Evaluation of Two High-Order Weighted Essentially Nonoscillatory Schemes," *AIAA Journal*, vol. 46, pp. 3090–3102, December 2008.

[44] Foster, N. F. and Noack, R. W., "High-Order Overset Interpolation Within an OVERFLOW Solution," in *50th AIAA Aerospace Sciences Meeting and Exhibit*, no. AIAA-2012-0728, (Nashville, Tennessee), January 9–12, 2012.

[45] Sherer, S. E. and Scott, J. N., "Comparison of Highly Accurate Interpolation Methods," in *39th AIAA Aerospace Sciences Meeting*, no. AIAA-2001-0282, (Reno, Nevada), January 8–11, 2001.

[46] Delfs, J. W., "An Overlapped Grid Technique for High Resolution CAA Schemes for Complex Geometries," in *7th AIAA/CEAS Aeroacoustics Conference*, no. AIAA-2001-2199, (Maastricht, Netherlands), May 28–30, 2001.

[47] Wendland, H., *Scattered Data Approximation*. New York, NY: Cambridge University Press, 2005.

[48] Whitehouse, G., Boschitsch, A., Quackenbush, T., Wachspress, D., and Brown, R. E., "Novel Eulerian Vorticity Transport Wake Module for Rotorcraft Flow Analysis," in *Proceedings of the 63rd American Helicopter Society Annual Forum*, (Virginia Beach, VA), May 1–3, 2007.

[49] Greengard, L. and Rokhlin, V., "A Fast Algorithm for Particle Simulations," *Journal of Computational Physics*, vol. 73, pp. 325–348, 1987.

[50] Carlson, J.-R., "Inflow/Outflow Boundary Conditions with Applications to FUN3D," Tech. Rep. NASA/TM-2011-217181, National Aeronautics and Space Administration, Langley Research Center, Hampton, Virginia, October 2011.

[51] Anderson, W. K. and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers & Fluids*, vol. 23, no. 1, pp. 1–21, 1994.

[52] Biedron, R., Vatsa, V., and Atkins, H., "Simulation of Unsteady Flows Using an Unstructured Navier-Stokes Solver on Moving and Stationary Grids," in *23rd AIAA Applied Aerodynamics Conference*, no. AIAA-2005-5093, (Toronto, Ontario), June 6–8, 2005.

[53] Sanchez-Rocha, M., Smith, M. J., and Menon, S., "Improved Prediction of Complex Rotorcraft Aerodynamics," in *American Helicopter Society 69th Annual Forum*, (Phoenix, AZ), May 21–23, 2013.

[54] Whitehouse, G. R. and Tadghighi, H., "Investigation of Hybrid Grid-Based Computational Fluid Dynamics Methods for Rotorcraft Flow Analysis," *Journal of the American Helicopter Society*, vol. 56, no. 032004, pp. 1–10, 2011.

[55] Whitehouse, G. R. and Boschitsch, A. H., "Toward the Next Generation of Grid-Based Vorticity-Velocity Solvers for General Rotorcraft Flow Analysis," in *Proceedings of the 69th American Helicopter Society Annual Forum*, (Phoenix, Arizona), May 21–23, 2013.

[56] Noack, R. W., Boger, D. A., Kunz, R. F., and Carrica, P. M., "Suggar++: An Improved General Overset Grid Assembly Capability," in *AIAA 19th Computational Fluid Dynamics Conference*, no. AIAA-2009-3992, (Austin, Texas), June 22-25, 2009.

[57] Fasshauer, G. E., "Positive Definite Kernels: Past, Present and Future," *Dolomites Research Notes on Approximation*, vol. 4, pp. 1–63, 2011.

[58] Quon, E. and Smith, M. J., "Advanced Interpolation Techniques for Overset CFD," in *50th AIAA Aerospace Sciences Meeting and Exhibit*, no. AIAA-2012-0305, (Nashville, Tennessee), January 9–12, 2012.

[59] Chaplot, V., Darboux, F., Bourennane, H., Leguedois, S., Silvera, N., and Phachomphon, K., "Accuracy of Interpolation Techniques for the Derivation of Digital Elevation Models in Relation to Landform Types and Data Density," *Geomorphology*, vol. 77, pp. 126–141, 2006.

[60] Fornberg, B. and Zuev, J., "The Runge Phenomenon and Spatially Variable Shape Parameters in RBF Interpolation," *Computers and Mathematics with Applications*, vol. 54, pp. 379–398, August 2007.

[61] Fornberg, B. and Flyer, N., *The Gibbs Phenomenon in Various Representations and Applications*, ch. Gibbs Phenomenon for Radial Basis Functions, pp. 201–224. Potsdam, New York: Sampling Publishing, 2011.

[62] Shu, C.-W., "Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws," Tech. Rep. NASA/CR-97-206253, NASA Langley Research Center, Hampton, Virginia, November 1997.

[63] Tam, C. K. and Kurbatskii, K. A., "A Wavenumber Based Extrapolation and Interpolation Method for Use in Conjunction with High-Order Finite Difference Schemes," *Journal of Computational Physics*, vol. 157, pp. 588–617, 2000.

[64] Desquesnes, G., Terracol, M., Manoha, E., and Sagaut, P., "On the Use of a High Order Overlapping Grid Method for Coupling in CFD/CAA," *Journal of Computational Physics*, vol. 220, pp. 355–382, 2006.

[65] Lee, K. R., Park, J. H., and Kim, K. H., "High-Order Interpolation Method for Overset Grid Based on Finite Volume Method," *AIAA Journal*, vol. 49, pp. 1387–1398, July 2011.

[66] Buhmann, M. D., *Radial Basis Functions*. New York, NY: Cambridge University Press, 2003.

[67] Lele, S. K., "Compact Finite Difference Schemes with Spectral-like Resolution," *Journal of Computational Physics*, vol. 103, pp. 16–42, 1992.

[68] Chicheportiche, J. and Gloerfelt, X., "Study of Interpolation Methods for High-Accuracy Computations on Overlapping Grids," *Computers and Fluids*, vol. 68, pp. 112–133, 2012.

[69] Kasson, J. M., "Tetrahedral Interpolation Algorithm Accuracy," in *Proceedings of SPIE–The International Society for Optical Engineering*, 1994.

[70] Hubbard, B. and Chen, H.-C., "A Chimera Scheme for Incompressible Viscous Flows with Application to Submarine Hydrodynamics," in *25th AIAA Fluid Dynamics Conference*, no. AIAA-94-2210, 1994.

[71] Bathe, K.-J., *Finite Element Procedures*. Upper Saddle River, New Jersey: Prentice Hall, 1996.

[72] Samareh, J. A., "Discrete Data Transfer Technique for Fluid-Structure Interaction," in *18th AIAA Computational Fluid Dynamics Conference*, no. AIAA-2007-4309, (Miami, Florida), June 25–28, 2007.

[73] Lee, P. K., "CFD/CSD Grid Interfacing of Three-Dimensional Surfaces by Inverse Isoparametric Mapping," Master's thesis, Massachusetts Institute of Technology, 2001. `http://dspace.mit.edu/handle/1721.1/81568`.

[74] Cook, R. D., Malkus, D. S., Plesha, M. E., and Witt, R. J., *Concepts and Applications of Finite Element Analysis*. John Wiley & Sons, fourth ed., 2002.

[75] Smith, M. J., Hodges, D. H., and Cesnik, C. E. S., "An Evaluation of Computational Algorithms to Interface Between CFD and CSD Methodologies," Tech. Rep. WL-TR-96-3055, Air Force Wright Laboratory, November 1995.

[76] Murti, V. and Valliappan, S., "Numerical Inverse Isoparametric Mapping in Remeshing and Nodal Quantity Contouring," *Computers and Structures*, vol. 22, no. 6, pp. 1011–1021, 1986.

[77] Smith, M. J., Hodges, D. H., and Cesnik, C. E. S., "An Evaluation of Computational Algorithms Suitable for Fluid-Structure Interaction," *AIAA Journal of Aircraft*, vol. 37, no. 2, pp. 282–294, 2000.

[78] Lee, N.-S. and Bathe, K.-J., "Effects of element distortions on the performance of isoparametric elements," *International Journal for Numerical Methods in Engineering*, vol. 36, pp. 3553–3576, 1993.

[79] Chen, J. and Li, C., "A Cubic Quadrliateral Spline Element with Concave Shapes," *Theoretical and Applied Mechanics Letters*, vol. 3, pp. 0310031–4, 2013.

[80] Cen, S., Fu, X.-R., and Zhou, M.-J., "8- and 12-node Plane Hybrid Stress-Function Elements Immune to Severely Distorted Mesh Containing Elements with Concave Shapes," *Computational Methods in Applied Mechanics and Engineering*, vol. 200, pp. 2321–2336, 2011.

[81] Shepard, D., "A Two-Dimensional Interpolation Function for Irregularly-Spaced Data," in *ACM National Conference*, 1968.

[82] Renka, R. J., "Multivariate Interpolation of Large Sets of Scattered Data," *Transactions on Mathematical Software*, vol. 14, no. 2, pp. 139–148, 1988.

[83] Forrester, A., Sobester, A., and Keane, A., *Engineering Design via Surrogate ModeModel: A Practical Guide*. John Wiley & Sons, 2008.

[84] Jarkowski, M., Woodgate, M., Rokicki, J., and Barakos, G., "Toward Consistent Hybrid Overset Mesh Methods for Rotorcraft CFD," in *Proceedings of the 37th European Rotorcraft Forum*, (Gallarate, Varese, Italy), 2011.

[85] Franke, R., "Scattered Data Interpolation: Tests of Some Methods," *Mathematics of Computations*, vol. 38, no. 157, pp. 181–200, 1982.

[86] Wood, J. D. and Fisher, P. F., "Assessing Interpolation Accuracy in Elevation Models," *IEEE Computer Graphics and Applications*, vol. 13, no. 2, pp. 48–56, 1993.

[87] Takeda, H., Farsiu, S., and Milanfar, P., "Kernel Regression for Image Processing and Reconstruction," *IEEE Transactions on Image Processing*, vol. 16, pp. 349–366, February 2007.

[88] Luo, J.-C., Leung, Y., Zheng, J., and Ma, J.-H., "An Elliptical Basis Function Network for Classification of Remote Sensing Images," *Journal of Geographical Systems*, vol. 6, pp. 219–236, 2004.

[89] Saloff-Coste, L., "The Heat Kernel and Its Estimates," in *Advanced Studies in Pure Mathematics*, vol. 57, pp. 405–436, Mathematical Society of Japan, 2010.

[90] Schaback, R., "Creating Surfaces from Scattered Data Using Radial Basis Functions," in *Mathematical Methods for Curves and Surfaces*, pp. 477–496, 1995.

[91] Smith, M. J., Cesnik, C. E. S., and Hodges, D. H., "Evaluation of Some Data Transfer Algorithms for Noncontiguous Meshes," *Journal of Aerospace Engineering*, vol. 13, no. 2, pp. 52–58, 2000.

[92] Bochner, S., "Lectures on Fourier Integrals," *Annals of Mathematical Studies*, 1959.

[93] Buhmann, M., "A New Class of Radial Basis Functions with Compact Support," *Mathematics of Computation*, vol. 70, no. 233, pp. 307–318, 2000.

[94] Beatson, R., Cherrie, J., and Mouat, C., "Fast Fitting of Radial Basis Functions: Methods Based on Preconditioned GMRES Iteration," *Advances in Computational Mathematics*, vol. 11, pp. 253–270, 1999.

[95] Micchelli, C. A., "Interpolation of Scattered Data; Distance Matrices and Conditionally Positive Definite Functions," *Constructive Approximation*, vol. 2, pp. 11–22, 1986.

[96] Lazzaro, D. and Montefusco, L. B., "Radial Basis Functions for the Multivariate Interpolation of Large Scattered Data Sets," *Journal of Computational and Applied Mathematics*, vol. 140, pp. 521–536, 2002.

[97] Wendland, H., *Topics in Multivariate Approximation and Interpolation*, ch. Computational Aspects of Radial Basis Function Approximation, pp. 231–256. Elsevier, 2005.

[98] Shonkwiler, R. W. and Lefton, L., *An Introduction to Parallel and Vector Scientific Computing*. New York, NY: Cambridge University Press, 2006.

[99] Rendall, T. and Allen, C., "Evaluation of Radial Basis Functions for CFD Volume Data Interpolation," in *48th AIAA Aerospace Sciences Meeting*, no. AIAA-2010-1414, (Orlando, Florida), January 4–7, 2010.

[100] Schaback, R., "Error Estimates and Condition Numbers for Radial Basis Function Interpolation," *Advances in Computational Mathematics*, vol. 3, pp. 251–264, 1995.

[101] Opfer, R., "Multiscale Kernels," *Advances in Computational Mathematics*, vol. 25, pp. 357–380, 2006.

[102] Gnoffo, P. A., "Global Series Solutions of Nonlinear Differential Equations with Shocks Using Walsh Functions," *Journal of Computational Physics*, vol. 258, pp. 650–688, February 2014.

[103] Casa, L. and Krueger, P., "Radial Basis Function Interpolation of Unstructured, Three-Dimensional, VoluVolume Particle Tracking Velocimetry Data," *Measurement Science and Technology*, vol. 24, no. 6, pp. 1–19, 2013.

[104] Mongillo, M., ""choosing basis functions and shape parameters for radial basis function methods"," *SIAM Undergraduate Research Online*, vol. 4, pp. 190–209, 2011.

[105] de Boer, A., van Zuijlen, A. H., and Bijl, H., *Advanced Computational Methods in Science and Engineering*, ch. Radial Basis Functions for Interface Interpolation and Mesh Deformation, pp. 143–178. 2010.

[106] Rendall, T. and Allen, C., "Efficient Mesh Motion Using Radial Basis Function with Data Reduction Algorithms," *Journal of Computational Physics*, vol. 228, no. 17, pp. 6231–2649, 2009.

[107] Costin, W. and Allen, C., "Numerical Study of Radial Basis Function Interpolation for Data Transfer Across Discontinuous Mesh Interfaces," *International Journal for Numerical Methods in Fluids*, vol. 72, pp. 1076–1095, 2013.

[108] Franke, R., "A Critical Comparison of Some Methods for Interpolation of Scattered Data," Tech. Rep. NPS-53-79-003, Naval Postgraduate School, March 1979.

[109] Zhang, Q., "High-Order, Multidimensional, and Conservative Coarse-Fine Interpolation for Adaptive Mesh Refinement," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, pp. 3159–3168, 2011.

[110] Shenoy, R., *Overset Adaptive Strategies for Complex Rotating Systems*. PhD thesis, Georgia Institute of Technology, Atlanta, Georgia, 2014. `https://smartech.gatech.edu/handle/1853/51796`.

[111] Florez, W., Bustamante, C., Giraldo, M., and Hill, A., "Local Mass Conservative Hermite Interpolation for the Solution of Flow Problems by a Multi-Domain Boundary Element Approach," *Applied Mathematics and Computation*, vol. 218, pp. 6446–6457, 2012.

[112] Song, G., Riddle, J., Fasshauer, G. E., and Hickernell, F. J., "Multivariate Interpolation with Increasingly Flat Radial Basis Functions of Finite Smoothness," *Advances in Computational Mathematics*, vol. 36, pp. 485–501, 2011.

[113] Cakmakci, O., Kaya, I., Fasshauer, G. E., Thompson, K. P., and Rolland, J. P., "Application of Radial Basis Functions to Represent Optical Freeform Surfaces," in *International Optical Design Conference*, SPIE, September 2, 2010.

[114] Takeda, H., van Beek, P., and Milanfar, P., "Spatio-Temporal Video Interpolation and Denoising Using Motion-Assisted Steering Kernel (Mask) Regression," in *IEEE International Conference on Image Processing*, (San Diego, California), October 2008.

[115] Seo, H. J. and Milanfar, P., "Training-Free, Generic Object Detection Using Locally Adaptive Regression Kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1688–1704, September 2010.

[116] Feng, X. and Milanfar, P., "Multiscale Principal Components Analysis for Image Local Orientation Estimation," in *Asilomar Conference on Signals Systems and Computers*, (Pacific Grove, California), pp. 478–482, IEEE, November 2002.

[117] Fundamentals of Probability: Lecture 15, "Multivariate Normal Distributions." MIT OpenCourseWare, October 29, 2008. http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-436j-fundamentals-of-probability-fall-2008/lecture-notes/MIT6_436JF08_lec15.pdf.

[118] Mavriplis, D., "Unstructured Grid Techniques," *Annual Review of Fluid Mechanics*, vol. 29, pp. 473–514, 1997.

[119] Baxter, B. J. C., *The Interpolation Theory of Radial Basis Functions*. PhD thesis, Trinity College, 1992. http://arxiv.org/pdf/1006.2443v1.

[120] Zhang, X., King, M. L., and Hyndman, R. J., "A Bayesian Approach to Bandwidth Selection for Multivariate Kernel Density Estimation," *Computational Statistics & Data Analysis*, vol. 50, no. 11, pp. 3009–3031, 2006.

[121] Mak, M.-W. and Kung, S.-Y., "Estimation of Elliptical Basis Function Parameters by the EM Algorithm with Application to Speaker Verification," *IEEE Transactions on Neural Networks*, vol. 11, pp. 961–969, 2000.

[122] Wu, Z. and Schaback, R., "Local Error Estimates for Radial Basis Function Interpolation of Scattered Data," *Journal of Numerical Analysis*, vol. 13, pp. 13–27, 1993.

[123] Wendland, H., "Error Estimates for Interpolation by Compact Supported Radial Basis Functions of Minimal Degree," *Journal of Approximation Theory*, vol. 93, pp. 258–272, 1998.

[124] Schaback, R., "Improved Error Bounds for Scattered Data Interpolation by Radial Basis Functions," *Mathematics of Computations*, vol. 68, no. 225, pp. 201–216, 1999.

[125] Visbal, M. R. and Gaitonde, D. V., "High-Order-Accurate Methods for Complex Unsteady Subsonic Flows," *AIAA Journal*, vol. 37, no. 10, pp. 1231–1239, 1999.

[126] Visbal, M. R. and Gaitonde, D. V., "On the Use of Higher-Order Finite-Difference Schemes on Curvilinear and Deforming Meshes," *Journal of Computational Physics*, vol. 181, pp. 155–185, 2002.

[127] John, J. E., *Gas Dynamics*. Upper Saddle River, New Jersey: Prentice Hall, 2nd ed., September 18 1997.

[128] van Leer, B., "Toward the Ultimate Conservative Difference Scheme. II. Monotonicity and Conservation Combined in a Second-Order Scheme," *Journal of Computational Physics*, vol. 14, pp. 361–370, 1974.

[129] Roper, D., Owen, I., Padfield, G., and Hodge, S., "Integrating CFD and Piloted Simulation to Quantify Ship-Helicopter Operating Limits," *The Aeronautical Journal*, vol. 110, no. 1109, pp. 419–428, 2006.

[130] Wilkinson, C., Zan, S., Gilbert, N., and Funk, J., "Modelling and Simulation of Ship Air Wakes for Helicpoter Operations–A Collaborative Venture," in *RTO AVT Symposium on "Fluid Dynamics Problems of Vehicles Operating near or in the Air-Sea Interface"*, (Amsterdam, The Netherlands), 1998.

[131] R. McKillip, J., Boschitsch, A., Quackenbush, T., Keller, J., and Wachspress, D., "Dynamic Interface Simulation Using a Coupled Vortex-Based Ship Airwake and Rotor Wake Model," in *Proceedings of the 58th American Helicopter Society Annual Forum*, (Montreal, Canada), June 11–13, 2002.

[132] Syms, G., "Simulation of Simplified-Frigate Airwakes Using a Lattice-Boltzmann Method," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 96, pp. 1197–1206, 2008.

[133] Wakefield, N., Newman, S., and Wilson, P., "Helicopter Flight Around a Ship's Superstructure," in *Institution of Mechanical Engineers Part G: Journal of Aerospace Engineering*, vol. 216, pp. 13–28, 2002.

[134] Cheney, B. and Zan, S., "CFD Code Validation Data and Flow Topology for the Technical Co-Operation Program AER-TP2 Simple Frigate Shape," Tech. Rep. NRC-IAR-LTR-A-035, National Research Council of Canada, Institute for Aerospace Research, 1999.

[135] Forrest, J. S. and Owen, I., "An Investigation of Ship Airwakes Using Detached-Eddy Simulation," *Computers & Fluids*, vol. 39, pp. 656–673, 2010.

[136] Quon, E., Smith, M. J., Rosenfield, N., and Whitehouse, G., "Investigation of Ship Airwakes Using a Hybrid Computational Methodology," in *Proceedings of the 70th American Helicopter Society Forum*, (Montreal, Canada), May 20–22, 2014.

[137] Anderson, W. K. and Bonhaus, D. L., "An Implict Unpwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, vol. 23, no. 1, pp. 1–21, 1994.

[138] Etienne, S., Garon, A., and Pelletier, D., "Code Verification for Unsteady Flow Simulation with High Order Time-Stepping Schemes," in *47th AIAA Aerospace Sciences Meeting*, no. AIAA-2009-0169, (Orlando, Florida), January 5–8, 2009.

[139] Anderson, W. K., "Grid Generation and Flow Solution Method for Euler Equations on Unstructured Grids," Tech. Rep. TM-4295, NASA, 1992.

[140] I. H. Abbott and A. E. von Doenhoff, *Theory of Wing Sections*. Mineola, NY: Dover Publications, 1959.

[141] Anderson, J. D., *Aircraft Performance & Design*. New York, NY: McGraw-Hill Science/Engineering/Math, 1998.

[142] Lynch, C. E., *Advanced CFD Methods for Wind Turbine Analysis*. PhD thesis, Georgia Institute of Technology, Atlanta, Georgia, 2011. `https://smartech.gatech.edu/handle/1853/39491`.

[143] F. X. Caradonna and C. Tung, "Experimental and Analytical Studies of a Model Helicopter Rotor in Hover," Tech. Rep. TM-81232, NASA, 1981.

[144] F.X. Caradonna and C. Tung, "Experimental and Analytical Studies of a Model Helicopter Rotor in Hover," *Vertica*, vol. 5, pp. 149–161.

[145] Shenoy, R., Smith, M. J., and Park, M. A., "Unstructured Overset Mesh Adaptation with Turbulence Modeling for Unsteady Aerodynamic Interactions," *AIAA Journal of Aircraft*, vol. 51, pp. 161–174, January 2014. doi: 10.2514/1.C032195.

[146] Thresher, R., Robinson, M., and Veers, P., "The Future of Wind Energy Technology in the United States," Tech. Rep. NREL/CP-500-43412, National Renewable Energy Laboratory, October 2008.

[147] Keck, R.-E., de Maré, M., Churchfield, M. J., Lee, S., Larsen, G., and Madsen, H. A., "Two Improvements to the Dynamic Wake Meandering Model: Inlcuding the Effects of Atmospheric Shear on Wake Turbulence and Incorporating Turbulence Build-up in a Row of Wind Turbines," *Wind Energy*, 2013.

[148] M. M. Hand, D. A. Simms, L. J. Fingersh, D. W. Jager, J. R. Cotrell, S. Schreck, and S. M. Larwood, "Unsteady Aerodynamics Experiment Phase VI: Wind Tunnel Test Configurations and Available Data Campaigns," Tech. Rep. TP-500-29955, National Renewable Energy Laboratory, December 2001.

[149] Masarati, P. and Sitaraman, J., "Coupled CFD/Multibody Analysis of NREL Unsteady Aerodynamic Experiment Phase VI Rotor," in *49th AIAA Aerospace Sciences Meeting*, no. AIAA-2011-153, (Orlando, Florida), January 4–7, 2011.

[150] Lienhard, J. H., "Synopsis of Lift, Drag, and Vortex Frequency Data for Rigid Circular Cylinders," Tech. Rep. Bulletin 300, Washington State University, 1966.

[151] Achenbach, E. and Heinecke, E., "On Vortex Shedding from Smooth and Rough Cylinders in the Range of Reynolds Numbers 6e3 to 5e6," *Journal of Fluid Mechanics*, vol. 109, pp. 239–251, 1981.

[152] Hoerner, S. F., *Fluid-Dynamic Drag: Theoretical, Experimental and Statistical Information*. Bakersfield, CA: Self, 1st ed., 1958.

[153] Vatsa, V. N. and Carpenter, M. H., "Higher Order Temporal Schemes with Error Control for Unsteady Navier-Stokes Equations," in *17th AIAA Computational Fluid Dynamics Conference*, no. AIAA-2005-5245, (Toronto, Canada), June 6–9, 2005.

[154] Churchfield, M. J., Laursen, J., Loeven, A., Lee, S., and Moriarty, P. J., "A Comparison Between Wind Turbine Aerodynamics Model Output When Using Generic Versus Actual Turbine Characterization as Input," in *51st AIAA Aerospace Sciences Meeting*, no. AIAA-2013-1208, (Grapevine, Texas), January 7–10, 2013.

[155] D. Simms, S. Schreck, M. Hand, and L.J. Fingersh, "NREL Unsteady Aerodynamics Experiment in the NASA-Ames Wind Tunnel: A Comparison of Predictions to Measurements," Tech. Rep. TP-500-29494, National Renewable Energy Laboratory, June 2001.

[156] Sørensen, J. N. and Kock, C. W., "A Model for Unsteady Rotor Aerodynamics," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 58, pp. 259–275, 1995.

[157] Mikkelsen, R., Sörensen, J., and Shen, W., "Modeling and Analysis of the Flow Field around a Coned Rotor," *Wind Energy*, vol. 4, no. 3, pp. 121–135, 2001.